

Universidade Estadual de Campinas
Instituto de Matemática, Estatística e Computação Científica -
Imecc
Departamento de Matemática Aplicada

**ESCOLHA ADIADA DO PARÂMETRO DE
PENALIZAÇÃO E DO TAMANHO DE PASSO
EM ALGORITMOS DE PONTOS INTERIORES**

Tese de Doutorado

Fernando Rocha Villas Bôas

Orientador: Prof. Dr. Clovis Perin Filho

CAMPINAS

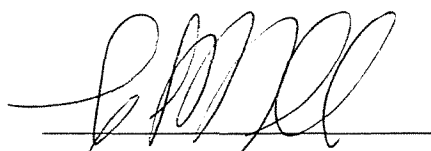
Fevereiro de 2000



**ESCOLHA ADIADA DO PARÂMETRO DE
PENALIZAÇÃO E DO TAMANHO DE PASSO
EM ALGORITMOS DE PONTOS INTERIORES**

Este exemplar corresponde à redação
final da tese devidamente corrigida
e defendida por
FERNANDO ROCHA VILLAS BÔAS
e aprovada pela comissão julgadora.

Campinas, 8 de fevereiro de 2000



Prof. Dr. Clovis Perin Filho
Orientador

Banca Examinadora:

- Prof. Dr. Clovis Perin Filho
- Prof. Dr. Horácio Hideki Yanasse
- Prof. Dr. Aurélio Ribeiro Leite de Oliveira
- Prof. Dr. José Mário Martínez Pérez
- Prof. Dr. Lúcio Tunes dos Santos

Tese apresentada ao Instituto de Matemática, Estatística e Computação Científica, UNICAMP, como requisito parcial para obtenção do Título de DOUTOR em Matemática Aplicada

UNIDADE	B. C.
N.º CHAMADA:	TI/UNICAMP
	V71e
V.	Ex.
TOMBO BC	40852
PROC.	278100
C	<input type="checkbox"/>
D	<input checked="" type="checkbox"/>
PREÇO	8,11,00
DATA	04/04/00
N.º CPD	00139

CM-00141270-1

**FICHA CATALOGRÁFICA ELABORADA PELA
BIBLIOTECA DO IMECC DA UNICAMP**

Villas Bôas, Fernando Rocha

V71e Escolha adiada do parâmetro de penalização e do tamanho de passo em algoritmos de pontos interiores / Fernando Rocha Villas Bôas -- Campinas, [S.P. :s.n.], 2000.

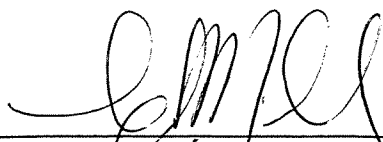
Orientador : Clovis Perin Filho

Tese (doutorado) - Universidade Estadual de Campinas, Instituto de Matemática, Estatística e Computação Científica.

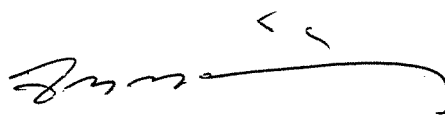
1. Programação linear. 2. Algoritmo. I. Perin Filho, Clovis. II. Universidade Estadual de Campinas. Instituto de Matemática, Estatística e Computação Científica. III. Título.

Tese de Doutorado defendida em 08 de fevereiro de 2000

e aprovada pela Banca Examinadora composta pelos Profs. Drs.



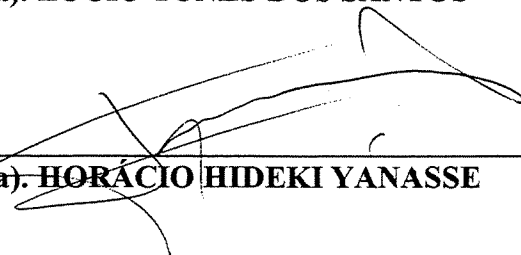
Prof (a). Dr (a). **CLOVIS PERIN FILHO**



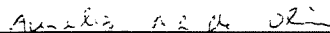
Prof (a). Dr (a). **JOSÉ MÁRIO MARTINEZ PEREZ**



Prof (a). Dr (a). **LÚCIO TUNES DOS SANTOS**



Prof (a). Dr (a). **HORÁCIO HIDEKI YANASSE**



Prof (a). Dr (a). **AURELIO RIBEIRO LEITE DE OLIVEIRA**

Agradecimentos

Ao Prof. Clovis Perin, pelos seus comentários e orientação ao longo desse trabalho.

Este trabalho foi realizado com o suporte financeiro da Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP), por meio da bolsa de doutoramento 95/6979-9.

Para Jo

Resumo

Nesse trabalho estudamos, no contexto de métodos de pontos interiores para programação linear, algumas possíveis vantagens de se adiar as escolhas do parâmetro de penalização e do tamanho de passo, que ocorrem tanto quando usamos o método de Newton para resolver o sistema de Karush-Kuhn-Tucker, como quando aplicamos um esquema preditor-corretor. Nós mostramos que, tanto para um passo de Newton quanto para um passo preditor-corretor, o próximo iterando pode ser expresso como uma função linear do parâmetro de penalização μ e, no caso de um passo preditor-corretor, como uma função quadrática de μ . Mostramos também que essa parametrização é útil para garantir, por exemplo, a não-negatividade do próximo iterando ou sua proximidade da trajetória central. Resultados computacionais dessas estratégias são apresentados e comparados com PCx, uma implementação do método preditor-corretor de Mehrotra.

Abstract

We study, in the context of interior-point methods for linear programming, some possible advantages of postponing the choice of the penalty parameter and the step length, which happens both when we apply Newton's method to the Karush-Kuhn-Tucker system and when we apply a predictor-corrector scheme. We show that for a Newton or a strictly predictor step the next iterate can be expressed as a linear function of the penalty parameter μ , and, in the case of a predictor-corrector step, as a quadratic function of μ . We also show that this parameterization is useful to guarantee either the non-negativity of the next iterate or the proximity to the central path. Computational results of these strategies are shown and compared with PCx, an implementation of Mehrotra's predictor-corrector method.

Índice

1	Introdução	1
1.1	Motivação histórica	1
1.2	Descrição do trabalho	5
2	Parametrização de primeira ordem em um contexto primal-dual	10
3	Parametrização de segunda ordem em um contexto primal-dual	15
4	Vizinhanças da trajetória central	19
5	Algoritmo	23
6	Resultados de convergência	25
7	Contexto auto-dual	34
8	Resultados computacionais	39
9	Conclusões	45
A	Parametrização no contexto auto-dual	47
A.1	Parametrização de primeira ordem	48
A.2	Parametrização de segunda ordem	61

Capítulo 1

Introdução

1.1 Motivação histórica

A programação linear trata de problemas de minimizar ou maximizar uma função linear na presença de restrições lineares. Desde o desenvolvimento do Método Simplex por George B. Dantzig em 1947, a programação linear tem sido usada extensivamente em setores militares, industriais, governamentais e de planejamento urbano entre outros. A popularidade da programação linear pode ser atribuída a muitos fatores, entre eles sua capacidade de modelar problemas grandes e complexos, e de possibilitar aos usuários resolver problemas grandes em um tempo razoável.

Entretanto, essa possibilidade de modelar e resolver problemas de grande porte nem sempre se mostrou disponível. A resolução de certos problemas grandes, originários de modelagens de problemas reais e importantes, não é satisfatória usando o método simplex.

Uma família de problemas que ilustra bem este ponto trata da determinação da melhor composição de diversos produtos, onde cada um deles pode ser feito escolhendo-se alguns dentre vários insumos possíveis. Se os produtos não têm nenhum insumo em comum o problema pode ser decomposto em problemas menores, e assim, problemas muito grandes

⁰Parte desse capítulo foi baseada em Gill *et al.* [8], Gonzaga [12], Bazarra e Jarvis [3], Goldfarb e Mehrotra [9] e Villas Bôas [35].

são resolvidos rapidamente. Entretanto, quando os diversos produtos têm vários insumos em comum, que são escassos, o problema passa a exigir um tempo muito grande para ser resolvido, e esse tempo aumenta exponencialmente à medida que aumenta o número de insumos em comum.

O tempo necessário para obter a solução desse tipo de problema é excessivo e ocorre em qualquer variante especializada do método simplex, de modo independente dos detalhes de implementação. Simplificadamente, o que ocorre é que os insumos em comum, usualmente chamados de restrições de acoplamento, estão presentes em quase todas as bases durante as iterações do método simplex, o que faz com que as inversas das bases fiquem densas.

Por esse motivo vários problemas tiveram que ser simplificados de modo insatisfatório para que pudessem ser resolvidos, o que gerou a motivação e a necessidade para que se prosseguisse na pesquisa de métodos com melhor desempenho para certos problemas grandes. Não teria havido motivação real para isso se, na prática, o desempenho do método simplex tivesse sido satisfatório para resolver todos os problemas grandes que surgiam.

Dessa motivação e necessidade surgiram os métodos de pontos interiores.

Muitos pesquisadores, começando pelo próprio Dantzig, observaram a característica aparentemente pouco eficiente do método simplex de caminhar ao longo da fronteira da região viável. Foram feitas tentativas de desenvolver métodos práticos de programação linear que cruzassem o interior da região viável — por exemplo Neuman [25], Hoffman *et al.* [13], Tompkins [32][33] e Frisch [7]. Tais métodos envolveram algumas vezes a aplicação de técnicas não-lineares à programação linear. Entretanto, nenhum desses métodos foi considerado, nem mesmo pelos próprios autores, como sendo competitivo em rapidez com o método simplex para a programação linear em geral.

Do lado teórico, a questão que se colocava era se existiria algum algoritmo polinomial para o problema da programação linear. Isso foi respondido em 1978 por Khachiyan [18], [19], que aplicou o método de Shor [29] e o método elipsóide de Yudin e Nemirovskii

[41] ao problema da programação linear, baseado na geometria não-linear de elipsóides que se reduzem, e demonstrou a existência de uma limitação superior ao número de operações aritméticas necessárias para se obter uma solução ótima. A limitação, $\mathcal{O}(n^4L)$, depende do número L , que é o número total de bits usados na descrição dos dados do problema. O método de Khachiyan, entretanto, mostrou-se não ser prático porque o número de iterações necessárias à obtenção da solução é quase sempre muito grande, próximo à limitação fornecida pela análise do pior caso, e porque os erros numéricos de arredondamento tendem a se acumular a cada iteração.

Em 1984, Karmarkar [17] apresentou um novo algoritmo com tempo polinomial para a programação linear. O método de Karmarkar se baseia em três pontos principais.

- Por meio de uma transformação projetiva de \mathbb{R}^n em \mathbb{R}^{n+1} , a região viável em \mathbb{R}^n (dada pela interseção de um espaço afim $Ax = b$ com o ortante positivo de \mathbb{R}^n) é transformada em uma região em \mathbb{R}^{n+1} , dada pela interseção de um subespaço vetorial (imagem do espaço afim através da transformação projetiva) com um simplex (imagem do ortante positivo de \mathbb{R}^n pela transformação projetiva). Essa transformação projetiva leva um ponto inicial do interior da região viável em \mathbb{R}^n ao centro do simplex de \mathbb{R}^{n+1} .
- A partir do centro do simplex e na direção da projeção do gradiente na região viável, é dado um passo cujo tamanho é limitado por uma função potencial que impede que o novo ponto fique próximo da fronteira.
- A cada iteração, um outro tipo de transformação projetiva, desta vez de \mathbb{R}^{n+1} em \mathbb{R}^{n+1} , transforma o problema da iteração interior, levando novamente o ponto que foi calculado antes para o centro do simplex.

Karmarkar demonstrou que seu método básico tem a mesma complexidade do método de Khachiyan, e apresentou também uma variante, usando *atualizações de posto um*, que tem complexidade $\mathcal{O}(n^{3.5}L)$.

Muito mais importante do que uma melhora na complexidade é o fato de que o método de Karmarkar forneceu um enfoque novo, original e prático para resolver problemas de programação linear. O método foi bastante divulgado pela mídia e nos meios acadêmicos, nem tanto por seus méritos teóricos — inegáveis —, mas mais pela alegação de que era capaz de resolver problemas reais de grande porte muito mais rapidamente do que o método simplex. Tentativas iniciais para verificar essa alegação confirmaram que o método de Karmarkar e métodos relacionados obtêm uma solução em um número muito pequeno de iterações. Esse número usualmente é menor que 50 e tende a crescer muito lentamente à medida que n , o tamanho do problema, cresce.

Entretanto, verificar as alegações a respeito do tempo computacional foi muito mais difícil. Os primeiros resultados publicados nessa direção foram os de Gill *et al.* [8] e Adler *et al.* [1], que indicaram que métodos baseados no enfoque de Karmarkar eram viáveis.

Essa aparente discrepância entre um pequeno número de iterações *versus* um grande tempo computacional, observada nas primeiras implementações, se deve ao fato de que a operação de maior complexidade nos métodos de ponto interior é a projeção a cada iteração de um vetor no núcleo de uma matriz, e a menos que sejam tomados cuidados especiais com relação à preservação da esparsidade, projetar no núcleo de uma matriz esparsa pode ser equivalente a projetar no núcleo de uma matriz densa.

O trabalho de Karmarkar suscitou de imediato uma reavaliação dos métodos não-lineares que já tinham sido tentados antes — por exemplo os métodos de função de penalização suave, que estavam então “fora de moda” devido ao mal-condicionamento que com frequência ocorre próximo a uma solução ótima —, e foi mostrada por Gill *et al.* [8] a conexão entre o trabalho de Karmarkar e o método da função de barreira logarítmica de Frisch [6], assim como foi discutida sua relação com o método de centros de Huard [14], por Gill *et al.* [8] e por Todd e Burrell [31].

Os métodos de pontos interiores são hoje bastante variados na sua essência e não é consensual ainda qual deles é o melhor, se é que isso de fato é possível. O trabalho de Gonzaga [12] é uma excelente apresentação de muitos desses métodos, especialmente

daqueles onde os pontos interiores caminham ao longo de uma trajetória (vide Megiddo [21]), apesar de seu trabalho não tratar das aproximações da trajetória central por séries de potências (ver a esse respeito Monteiro *et al.* [23]).

1.2 Descrição do trabalho

Vários métodos do tipo primal-dual para a programação linear usam a função de barreira logarítmica para gerar uma família de problemas penalizados por um parâmetro μ ; cada problema é resolvido aproximadamente e o parâmetro de penalização μ é reduzido a cada iteração, formando uma seqüência que converge para zero. Alguns desses métodos têm que usar também um tamanho de passo α devido à restrição de não-negatividade. Esse enfoque é atribuído a Frisch [6] e é estudado em Fiacco & McCormick [5] no contexto de otimização não-linear, mas foi considerado pela primeira vez para a programação linear por Gill *et al.* [8].

Em implementações práticas, tais como a do preditor-corretor de Mehrotra [22] no PCx [4], as escolhas de parâmetro de penalização e do tamanho de passo são feitas usando heurísticas que são muito eficazes na prática — mas nós não dispomos ainda de ferramentas analíticas para compreender porque essas heurísticas funcionam tão bem.

Nesse trabalho propomos estabelecer um ambiente analítico que nos permita enunciar as escolhas do parâmetro de penalização e do tamanho de passo como um claro sub-problema de otimização. Isso é feito expressando o próximo iterando como uma função quadrática de μ para o método preditor-corretor ou como uma função linear para um passo de Newton, e depois estendendo para passos de Newton truncados.

De um modo mais geral, quaisquer que sejam as *equações de passo* usadas, isto é, sejam elas as equações de Newton para o próximo passo ou qualquer uma das diferentes versões das equações para o passo preditor-corretor, é sempre possível parametrizar o próximo iterando em termos de μ e do tamanho de passo α como uma função polinomial de μ e α , e isso ocorre porque essas equações são lineares.

Entretanto, computar os vetores que formam essa parametrização do próximo iterando tem conseqüências positivas e negativas.

Como conseqüência positiva, ser capaz de prever como será o próximo iterando, por meio de uma fórmula envolvendo apenas vetores conhecidos e sem ter que escolher antecipadamente os valores de μ e α , nos permite adiar essa escolha e focalizar nossa atenção nos critérios que um algoritmo em particular exige do próximo iterando para que haja convergência, como, por exemplo, os critérios de não-negatividade ou de proximidade da trajetória central. Tendo enunciado explicitamente em que consiste a garantia de que o próximo iterando satisfaz tal critério de convergência, sem ter estabelecido ainda os valores de μ e α , podemos então escolher valores “ótimos” para esses parâmetros, no sentido de garantir convergência rápida. Tal escolha adiada “ótima” depende de se minimizar localmente uma função adequada do próximo iterando, tal como o *gap dual* ou a *complementaridade* $x^T z$ da solução primal-dual corrente.

Como conseqüência negativa, temos que efetuar resoluções de sistema adicionais para cada vetor que multiplica uma potência de μ . Por exemplo, em algoritmos primais-duais sem passo corretor, após obtermos o fator de Cholesky é necessário resolver apenas um sistema linear depois de ter escolhido μ , enquanto que se essa escolha for adiada temos que resolver dois sistemas. Como outro exemplo, em um algoritmo preditor-corretor padrão é necessário resolver dois sistemas para cada fator de Cholesky — mas em nosso caso de escolha adiada temos que resolver três sistemas. Essa dificuldade é estrutural e inerente à escolha adiada de μ .

Mais do que isso, apesar de estarmos usando o mesmo fator de Cholesky, o esforço adicional de cada resolução extra de sistemas *não* é desprezível. Em uma amostragem típica de problemas da Netlib que avaliamos, o tempo necessário para resolver um sistema positivo definido, já dispondo do fator de Cholesky, é em média 20% do tempo necessário para computar o próprio fator de Cholesky e essa média tem uma variância alta. Esse resultado vai contra nossa intuição que se baseia em problemas densos, onde esses tempos têm ordem de grandeza menores do que o tempo para computar um fator de Cholesky.

Os fatos acima impõem *a priori* um limite para o que poderíamos esperar de implementações que adiam a escolha de μ e α — só podemos esperar uma redução dos tempos de CPU se a redução no número de iterações for suficientemente grande quando comparada com um algoritmo de referência.

Entretanto, o que queremos mostrar mais que tudo nesse trabalho é que, para um dado sistema de equações de passo baseado na função de barreira logarítmica, a parametrização do próximo iterando em termos de μ e α nos fornece os meios de avaliar e melhorar as demais partes do dado algoritmo.

Faremos isso da seguinte maneira. Primeiro mostraremos como a parametrização pode ser feita explicitamente em alguns exemplos de complexidade crescente. Para isso desenvolveremos as expressões dessa parametrização em um contexto primal-dual padrão e depois as estenderemos para incluir passos corretores, supondo em ambos os casos que um ponto inicial factível está disponível.

Depois mostraremos como essa parametrização pode ser utilizada. Uma vez que a proximidade da trajetória central é um dos poucos critérios disponíveis para garantir a convergência, nós o escolhemos como critério principal que o próximo iterando deve satisfazer e, usando diferentes critérios de proximidade, geramos uma família de algoritmos com diferentes complexidades polinomiais.

Em seguida estendemos essas idéias para um contexto auto-dual a fim de relaxar a necessidade de dispor de um ponto inicial factível.

Os algoritmos auto-duais foram estudados originalmente por Goldman e Tucker [10] e Tucker [34] e foram redescobertos por Ye *et al.* [39] no contexto de métodos de pontos interiores. Uma implementação inicial por Xu *et al.* [37] apresentou resultados promissores para problemas da Netlib, especialmente para uma nova classe de problemas infactíveis. Outros trabalhos em algoritmos auto-duais são os de Xu [36], Xu e Ye [38], e Hung e Ye [15].

Escolhemos efetuar as iterações em um espaço auto-dual para ficarmos livres de questões relativas à factibilidade — em um contexto auto-dual, dada uma solução inicial

não factível, podemos sempre gerar um problema que em certo sentido é equivalente ao programa linear original, e onde todos os iterandos são factíveis. Esse problema sempre tem uma solução ótima finita, que podemos usar para encontrar a solução do problema original ou para declarar que o problema é infactível ou tem solução ilimitada.

Desenvolvemos as expressões da parametrização em um contexto primal-dual e supondo que um ponto inicial factível está disponível. Fazemos isso a fim de apresentar nossas idéias de um modo mais simples e claro. Para desenvolver as expressões que foram utilizadas na implementação, nós usamos o contexto auto-dual com variáveis canalizadas. Nesse caso a notação se torna mais pesada mas as expressões estão explicitadas no apêndice A.

A fim de verificar a utilidade prática dos vários algoritmos baseados na parametrização proposta, utilizamos a biblioteca do PCx, isto é, utilizamos da implementação original as mesmas rotinas para entrada e saída, álgebra linear, pré-condicionamento, ponto inicial e critério de parada. Dessa forma todas as comparações resultantes são justas e têm significado. Para cada tipo de vizinhança usado fizemos uma implementação diferente, de modo que os resultados computacionais estão disponíveis para comparação.

Os resultados computacionais mostram que conseguimos uma redução no número de iterações, apesar de os tempos computacionais serem maiores.

Acreditamos que esses resultados são positivos. Apesar de os tempos de CPU serem maiores, a redução no número de iterações mostrou que a parametrização permitiu de fato uma visão mais clara sobre quais são os sub-problemas que têm que ser resolvidos a cada iteração.

Além disso, o fato de trabalharmos no espaço auto-dual faz com que sejam necessárias sete resoluções de sistemas lineares para cada fator de Cholesky em um algoritmo preditor-corretor, o que representa cinco resoluções extras se compararmos, por exemplo, com o algoritmo preditor-corretor de Mehrotra. Com isso, implementações práticas usando o contexto auto-dual tornam-se proibitivas para nossa parametrização.

Esse trabalho está organizado como segue. No Capítulo 2 desenvolvemos a parame-

trização linear para um passo estritamente de Newton. No Capítulo 3 estendemos essa idéia para um passo preditor-corretor truncado. Apresentamos uma visão das várias vizinhanças da trajetória central no Capítulo 4 e nosso algoritmo geral de trajetória central no Capítulo 5. Em seguida mostramos alguns resultados de convergência no Capítulo 6 e apresentamos o contexto auto-dual no Capítulo 7. Os resultados computacionais são apresentados no Capítulo 8.

Capítulo 2

Parametrização de primeira ordem em um contexto primal-dual

Consideramos o programa linear padrão e seu dual

$$(P) \min \{c^T x : Ax = b, x \geq 0\}$$

e

$$(D) \max \{b^T y : A^T y + z = c, z \geq 0\},$$

onde A é uma matriz $m \times n$ e b e c são vetores de dimensão m e n respectivamente. Como é usual em Programação Linear, a notação $x \geq 0$ ou $x > 0$, quando x é um vetor de dimensão n , indica a relação componente a componente, i.e., $x \geq 0 \Leftrightarrow x_i \geq 0, i = 1, \dots, n$.

Supomos que os conjuntos

$$\mathcal{F}_P = \{x \in \mathbb{R}^n \mid Ax = b, x > 0\}$$

e

$$\mathcal{F}_D = \{(y, z) \in \mathbb{R}^m \times \mathbb{R}^n \mid A^T y + z = c, z > 0\}$$

são ambos não vazios e definimos

$$\mathcal{F} = \mathcal{F}_P \times \mathcal{F}_D.$$

Resolveremos (P) usando a técnica da função de barreira logarítmica, que consiste em resolver aproximadamente a família de problemas

$$(P_\mu) : \min \left\{ c^T x - \mu \sum_{j=1}^n \ln x_j : Ax = b, x > 0 \right\},$$

onde $\mu > 0$ é o parâmetro de penalização e deve ser escolhido de modo que forme uma seqüência que converge para zero. Para cada μ o problema P_μ tem uma função objetivo que é estritamente convexa, de modo que o ótimo global é completamente caracterizado pelas condições de Karush-Kuhn-Tucker (ponto estacionário)

$$\begin{aligned} Ax - b &= 0 \\ A^T y + z - c &= 0 \\ xz - \mu e &= 0 \\ x > 0, \quad z > 0, \end{aligned} \tag{2.1}$$

onde e é o vetor de dimensão n que consiste apenas de uns, e adotamos a convenção de que se x e z são vetores de mesma dimensão então xz denota o vetor cujas componentes são $x_i z_i$. O produto escalar será denotado por $x^T z$.

A notação acima, xz , para denotar o produto componente a componente de dois vetores x e z de mesma dimensão, tem se estabelecido em trabalhos mais recentes em pontos interiores, tais como o de Jansen *et al.* [16] e os de Peng *et al.* [27][28], devido à frequência com que esse conceito é usado e para simplificar a notação. Da mesma maneira, se x é um vetor e a um escalar, denotamos por x^a o vetor cujas componentes são $(x_i)^a$.

A fim de resolver aproximadamente cada problema P_μ , e definindo $w \equiv (x, y, z)$

resolvemos, também aproximadamente, o sistema não-linear

$$H(w) = H(x, y, z) = \begin{pmatrix} Ax - b \\ A^T y + z - c \\ xz - \mu e \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}, \quad (2.2)$$

usando uma iteração do método de Newton, para o qual precisamos do Jacobiano de H :

$$J = \begin{pmatrix} A & 0 & 0 \\ 0 & A^T & I \\ Z & 0 & X \end{pmatrix},$$

onde X e Z são matrizes diagonais cujos elementos são iguais a x e z , respectivamente. De modo análogo ao que fizemos com vetores, estendemos a notação de potência para matrizes diagonais: se X é uma matriz diagonal e a um escalar definimos X^a como sendo a matriz diagonal cujos elementos não nulos são $(x_{i,i})^a$.

Estamos supondo que o ponto atual (x, y, z) é um ponto interior factível, i.e.

$$\begin{aligned} Ax &= b \\ A^T y + z &= c \end{aligned}$$

$$x > 0, z > 0.$$

A direção de Newton $\delta w = (\delta x, \delta y, \delta z)$ que fornece a solução aproximada $\hat{w} = (\hat{x}, \hat{y}, \hat{z}) = (x + \delta x, y + \delta y, z + \delta z)$ para o sistema $H(w) = 0$ é dada pela solução do

sistema linear $J(w) \delta w = -H(w)$, i.e.,

$$\begin{cases} A\delta x = 0, \\ A^T \delta y + \delta z = 0, \\ Z\delta x + X\delta z = \mu e - xz. \end{cases} \quad (2.3)$$

Definindo as matrizes

$$\begin{cases} \mathcal{P}_x = Z^{-1} - Z^{-1}XA^T(AZ^{-1}XA^T)^{-1}AZ^{-1}, \\ \mathcal{P}_y = (AZ^{-1}XA^T)^{-1}AZ^{-1}, \\ \mathcal{P}_z = A^T(AZ^{-1}XA^T)^{-1}AZ^{-1}, \end{cases}$$

obtemos por cálculo direto

$$\begin{aligned} \delta x &= \mathcal{P}_x(\mu e - xz), \\ \delta y &= -\mathcal{P}_y(\mu e - xz), \\ \delta z &= \mathcal{P}_z(\mu e - xz), \end{aligned} \quad (2.4)$$

Se agora definirmos

$$\begin{aligned} \gamma_{x0} &= -\mathcal{P}_x xz \\ \gamma_{x1} &= \mathcal{P}_x e \\ \gamma_{y0} &= \mathcal{P}_y xz \\ \gamma_{y1} &= -\mathcal{P}_y e \\ \gamma_{z0} &= \mathcal{P}_z xz \\ \gamma_{z1} &= -\mathcal{P}_z e \end{aligned} \quad (2.5)$$

temos então

$$\begin{aligned}\delta x &= \gamma_{x1}\mu + \gamma_{x0} \\ \delta y &= \gamma_{y1}\mu + \gamma_{y0} \\ \delta z &= \gamma_{z1}\mu + \gamma_{z0}\end{aligned}\tag{2.6}$$

i.e., expressamos cada novo passo (e portanto cada novo iterando) como uma função linear de μ . Observe que os vetores γ podem ser calculados sem atribuir nenhum valor a μ .

Em termos geométricos, a matriz diagonal $D = X^{1/2}Z^{-1/2}$, é uma matriz de *scaling*, $Z^{1/2}X^{-1/2}\mathcal{P}_x Z^{1/2}X^{1/2}$ é a matriz de projeção no espaço nulo de AD , $Z^{1/2}X^{-1/2}\mathcal{P}_z Z^{1/2}X^{1/2}$ é a matriz de projeção no espaço ortogonal de AD e $\mathcal{P}_y Z^{1/2}X^{1/2}$ fornece os multiplicadores de Lagrange que descrevem $(\mu e - xz)$ como uma combinação linear das linhas de AD .

Capítulo 3

Parametrização de segunda ordem em um contexto primal-dual

Suponhamos que estamos tentando determinar um único passo tal que

$$\hat{w} = (\hat{x}, \hat{y}, \hat{z}) = (x + \Delta x, y + \Delta y, z + \Delta z),$$

resolva exatamente o sistema não-linear (2.2). Por substituição direta podemos ver que tal passo tem que satisfazer o sistema não-linear

$$\begin{cases} A\Delta x = 0 \\ A^T \Delta y + \Delta z = 0 \\ Z\Delta x + X\Delta z = \mu e - xz - \Delta x \Delta z \end{cases}$$

A idéia por detrás de um passo de segunda ordem ou passo preditor-corretor é a de considerar que o termo não-linear $\Delta x \Delta z$ pode ser aproximado por $\delta x \delta z$ onde δx e δz são a solução do sistema linear (2.3) descrito no capítulo anterior (como veremos no Teorema 3:, próximo da trajetória central temos $\|\Delta x \Delta z - \delta x \delta z\| \leq \theta \mu$, i.e., $\Delta x \Delta z$ é próximo de $\delta x \delta z$ e isso vai de encontro à nossa idéia intuitiva de aproximação).

Com isso o sistema agora se torna linear:

$$\begin{cases} A\Delta x = 0 \\ A^T\Delta y + \Delta z = 0 \\ Z\Delta x + X\Delta z = \mu e - xz - \delta x\delta z \end{cases} \quad (3.1)$$

Por cálculo direto obtemos

$$\begin{aligned} \Delta x &= \mathcal{P}_x (\mu e - xz - \delta x\delta z) \\ \Delta y &= -\mathcal{P}_y (\mu e - xz - \delta x\delta z) \\ \Delta z &= \mathcal{P}_z (\mu e - xz - \delta x\delta z) \end{aligned} \quad (3.2)$$

Mas, de acordo com (2.6), podemos expressar δx e δz como uma função linear de μ e, assim, temos $\mu e - xz - \delta x\delta z = \mu e - xz - (\gamma_{x1}\mu + \gamma_{x0})(\gamma_{z1}\mu + \gamma_{z0}) = -\gamma_{x1}\gamma_{z1}\mu^2 + (e - \gamma_{x0}\gamma_{z1} - \gamma_{x1}\gamma_{z0})\mu - xz - \gamma_{x0}\gamma_{z0}$.

Definimos agora

$$\begin{aligned} \psi_2^x &= -\mathcal{P}_x \gamma_{x1}\gamma_{z1} \\ \psi_1^x &= \mathcal{P}_x (e - \gamma_{x0}\gamma_{z1} - \gamma_{x1}\gamma_{z0}) \\ \psi_0^x &= -\mathcal{P}_x (xz + \gamma_{x0}\gamma_{z0}) \\ \psi_2^y &= \mathcal{P}_y \gamma_{x1}\gamma_{z1} \\ \psi_1^y &= \mathcal{P}_y (\gamma_{x0}\gamma_{z1} + \gamma_{x1}\gamma_{z0} - e) \\ \psi_0^y &= \mathcal{P}_y (xz + \gamma_{x0}\gamma_{z0}) \\ \psi_2^z &= -\mathcal{P}_y \gamma_{x1}\gamma_{z1} \\ \psi_1^z &= \mathcal{P}_z (e - \gamma_{x0}\gamma_{z1} - \gamma_{x1}\gamma_{z0}) \\ \psi_0^z &= -\mathcal{P}_z (xz + \gamma_{x0}\gamma_{z0}) \end{aligned} \quad (3.3)$$

de modo que o passo resultante pode ser escrito como uma função quadrática de μ

$$\begin{aligned}\Delta x &= \psi_2^x \mu^2 + \psi_1^x \mu + \psi_0^x, \\ \Delta y &= \psi_2^y \mu^2 + \psi_1^y \mu + \psi_0^y, \\ \Delta z &= \psi_2^z \mu^2 + \psi_1^z \mu + \psi_0^z.\end{aligned}$$

No caso de um passo truncado, o próximo iterando \hat{w} pode ser expresso como uma função de α e μ já que

$$\begin{aligned}\hat{w}(\alpha, \mu) &= (x + \alpha\Delta x, y + \alpha\Delta y, z + \alpha\Delta z) \\ &= \begin{bmatrix} x + \alpha(\psi_2^x \mu^2 + \psi_1^x \mu + \psi_0^x) \\ y + \alpha(\psi_2^y \mu^2 + \psi_1^y \mu + \psi_0^y) \\ z + \alpha(\psi_2^z \mu^2 + \psi_1^z \mu + \psi_0^z) \end{bmatrix}\end{aligned}\tag{3.4}$$

O próximo teorema fornece a expressão do *gap dual* e será necessário para o algoritmo. Antes precisamos de um lema simples.

Lema 1: *Os vetores δx e δz são ortogonais; os vetores Δx e Δz são ortogonais.*

Demonstração: Reproduzimos a demonstração de [24]. δx e δz satisfazem o sistema (2.3). Multiplique a primeira equação por δy^T e a segunda por δx e as combine para obter $\delta x^T \delta z = 0$. Use a mesma idéia no sistema (3.1).

Teorema 1: *Seja \hat{g} o gap dual após um passo truncado de segunda ordem e seja g o gap anterior. Então $\hat{g} = \alpha n \mu + (1 - \alpha) g$.*

Demonstração:

$$\begin{aligned}\hat{g} &= \sum_{i=1}^n \hat{x}_i \hat{z}_i = \sum_{i=1}^n (x_i + \alpha \Delta x_i) (z_i + \alpha \Delta z_i) \\ &= \sum_{i=1}^n x_i z_i + \alpha \sum_{i=1}^n (x_i \Delta z_i + z_i \Delta x_i) + \alpha^2 \sum_{i=1}^n \Delta x_i \Delta z_i \\ &= g + \alpha \sum_{i=1}^n (\mu - x_i z_i - \delta x_i \delta z_i) = g + \alpha (n\mu - g).\end{aligned}$$

Observe que o mesmo teorema vale para passos de primeira ordem.

Capítulo 4

Vizinhanças da trajetória central

A trajetória central é definida como

$$\mathcal{C} = \{(x, y, z) \in \mathcal{F} : xz = \mu e\},$$

i.e., ela é uma curva contínua parametrizada pela variável real μ — para cada μ o ponto da curva é completamente caracterizado como sendo a solução única do sistema (2.1). Ela foi estudada por Bayer e Lagarias [2], Sonnevend [30] e Megiddo [21], entre outros. Uma vez que nas implementações de métodos de pontos interiores (IPM) nem sempre o vetor y é necessário ao longo das iterações, nós o omitiremos em alguns contextos e consideraremos apenas o par (x, z) .

Citando Hung e Ye [15], algoritmos que seguem a trajetória central geram uma sequência de pontos dentro de uma certa vizinhança da trajetória central \mathcal{C} , a qual impede os iterandos de se aproximarem muito prematuramente da fronteira da região viável.

Seja agora (x, z) um dado iterando. Esse iterando é a solução de um sistema linear envolvendo um dado μ , e assim existe um único ponto na trajetória central caracterizado por μ , o qual chamamos de μ -centro. A fim de definir uma vizinhança temos que ser capazes de quantificar a distância do vetor xz ao μ -centro. Seguindo as idéias de Peng *et al.* [27], essa quantificação pode ser feita definindo-se as seguintes medidas de proximidade.

$$\begin{aligned}\delta_F(xz, \mu) &= \sum_{i=1}^n \left(\frac{x_i z_i}{\mu} - 1 - \log \left(\frac{x_i z_i}{\mu} \right) \right), \\ \delta_K(xz, \mu) &= \left\| \frac{xz}{\mu} - e \right\|, \\ \delta_J(xz, \mu) &= \frac{1}{2} \left\| \sqrt{\frac{xz}{\mu}} - \sqrt{\frac{\mu}{xz}} \right\|.\end{aligned}$$

A menos que seja especificado de modo diferente, $\|\cdot\|$ denota sempre a norma euclidiana. De modo similar às definições anteriores, $\sqrt{\frac{\mu}{xz}}$ denota o vetor cujas componentes são $\sqrt{\frac{\mu}{x_i z_i}}$.

Em δ_J a raiz quadrada do vetor deve ser entendida como sendo o vetor cujas componentes são as raízes quadradas de cada componente. Observe que para todas as três medidas o seu valor é zero se xz for o ponto na trajetória central.

A primeira medida, conhecida como *função de barreira logarítmica* com respeito ao *parâmetro de barreira* μ , foi introduzida por Frisch [6] e historicamente foi a primeira a aparecer. Apesar de não ser prática de se implementar, ela é útil na análise de complexidade de vários IPMs, devido ao fato de ter a *propriedade de barreira*: ela se torna ilimitada quando xz se aproxima da fronteira do ortante positivo. Essa medida é particularmente útil na análise de algoritmos que fazem uma atualização drástica de μ e em seguida dão vários passos truncados para centrar.

A segunda medida δ_K foi introduzida por Kojima *et al.* [20] e é a mais usada para análise de complexidade. Entretanto ela tem a desvantagem de não ter a propriedade de barreira.

A terceira é devida a Jansen *et al.* [16] e parece ser capaz de produzir demonstrações mais concisas de complexidade, incluindo a de algoritmos de passo truncado. A motivação dessa medida, além de ter também a propriedade de barreira, é a de trabalhar em um espaço escalonado, onde as notações e demonstrações se tornam mais simples e onde essa medida corresponde a uma medida euclidiana clássica. Trabalhos mais recentes

de Peng *et al.* [28] propõem novos algoritmos baseados nessa medida que podem vir a ter seu desempenho prático confirmado — mas, apesar de promissora, ela ainda não foi capaz de produzir resultados revolucionários quanto à complexidade ou praticidade de implementação.

Focalizaremos apenas as medidas δ_K e δ_J e, para fins de implementação, usaremos apenas δ_K .

Podemos agora definir as vizinhanças, usando inicialmente a norma euclidiana e depois estendendo a outras normas.

Definiremos agora as vizinhanças \mathcal{N} da trajetória central para os vários critérios de proximidade δ .

Associada à medida de proximidade δ_K temos a definição clássica

$$\mathcal{N}_2(\beta) = \left\{ (x, y, z) \in \mathcal{F} : \left\| \frac{xz}{\mu} - e \right\| \leq 1 - \beta \right\},$$

onde $\beta \in (0, 1)$. Essa definição tem outra forma equivalente

$$\mathcal{N}_2(\beta) = \{ (x, y, z) \in \mathcal{F} : \|xz - \mu e\| \leq (1 - \beta)\mu \},$$

e usaremos livremente qualquer uma das formas conforme for mais conveniente.

A segunda vizinhança, associada a δ_J , é definida como

$$\mathcal{N}_J(\beta) = \left\{ (x, y, z) \in \mathcal{F} : \left\| \sqrt{\frac{xz}{\mu}} - \sqrt{\frac{\mu}{xz}} \right\| \leq \beta \right\}$$

onde $\beta > 0$.

A primeira vizinhança tem duas extensões naturais, $\mathcal{N}_\infty(\beta)$ onde a definição é a mesma, mas a norma é l_∞ , e

$$\mathcal{N}_\infty^-(\beta) = \{ (x, y, z) \in \mathcal{F} : xz \geq \beta\mu \}.$$

Podemos ver que

$$\mathcal{C} \subset \mathcal{N}_2(\beta) \subset \mathcal{N}_\infty(\beta) \subset \mathcal{N}_\infty^-(\beta) \subset \mathcal{F}$$

A extensão para a segunda vizinhança é similar.

Dizemos que um ponto está próximo da trajetória central sob um determinado critério de proximidade \mathcal{N} se ele pertence a uma das vizinhanças acima para um dado $\beta > 0$.

Podemos agora enunciar um algoritmo geral de trajetória central. Entretanto, não estamos simplesmente buscando generalidade. Em vez disso, restringiremos o algoritmo geral para o “menor” contexto que englobe todos os algoritmos que de fato implementamos e para os quais dispomos de resultados computacionais.

Capítulo 5

Algoritmo

Apresentamos agora um esquema de algoritmo que engloba todos as implementações que fizemos, no sentido que ele permite que se considere passos de primeira ou segunda ordem, e vários tipos diferentes de vizinhança.

Passo 0) Suponha que seja dado um critério de proximidade \mathcal{N} e seja $w_0 = (x_0, y_0, z_0)$ um ponto inicial factível próximo da trajetória central sob esse critério de proximidade. Seja $\varepsilon > 0$ a tolerância para o gap dual.

Passo 1) Se $g_k = x_k^T z_k \leq \varepsilon$ pare.

Passo 2) Compute a função $\Delta w_k(\mu)$, i.e., compute de acordo com (2.5) os vetores γ que permitem que o novo passo Δw_k possa ser expresso como uma função de μ (ou no caso de uma parametrização de segunda ordem compute de acordo com (3.3) os vetores ψ).

Passo 3) Minimize (aproximadamente) em α e μ a função $g_{k+1}(\alpha, \mu) = \alpha n \mu + (1 - \alpha) g_k$ sujeito a $w_{k+1}(\alpha, \mu) = w_k + \alpha \Delta w_k(\mu)$ ficar próximo da trajetória central.

Passo 4) Faça $w_{k+1} = w_k + \alpha \Delta w_k$ e $k \leftarrow k + 1$ e vá para passo 1.

Para mostrar que o algoritmo acima está bem definido temos que mostrar que ele termina adequadamente e que todos os passos podem ser executados. Não há problema

quanto aos passos 1 e 4. Para o Passo 0 o problema de encontrar um ponto inicial próximo da trajetória central será tratado quando estendermos nossas idéias para um contexto auto-dual. Os Capítulos 2 e 3 mostraram que o Passo 2 não apresenta problemas. O Passo 3 origina por si só um subproblema não-linear que está bem posto — veremos que podemos enunciar e resolver esse problema aproximadamente (ou mesmo exatamente para alguns critérios de proximidade), desde que o conjunto solução associado às restrições não seja vazio. Mostraremos que esse é o caso dependendo do critério escolhido.

Também dependendo do critério de proximidade haverá diferentes complexidades polinomiais associadas. No capítulo seguinte elaboraremos mais sobre o conjunto restrição para o Passo 3 e sobre a complexidade polinomial.

Capítulo 6

Resultados de convergência

Os resultados apresentados neste capítulo garantem a convergência do algoritmo do capítulo anterior para as vizinhanças $\mathcal{N}_2(\beta)$ e $\mathcal{N}_\infty^-(\beta)$, para valores específicos de β que incluem os valores usados nas implementações. Observe que a complexidade do algoritmo é mantida polinomial.

Adotaremos a convenção de que se x é um ponto em uma iteração então \hat{x} denota o próximo iterando.

Teorema 2: *Seja \mathcal{N} o critério do algoritmo definido por $\beta \in (0, 1/2)$ e vizinhança $\mathcal{N}_2(\beta)$ e suponha que estamos usando apenas passos de primeira ordem. Então o Passo 3 do algoritmo do Capítulo 5 admite uma minimização aproximada. Além disso o algoritmo pára em $\mathcal{O}(\sqrt{n}L)$ passos.*

Demonstração: Esse é o algoritmo primal-dual clássico de Monteiro e Adler e sua demonstração pode ser encontrada em [24].

A fim de demonstrar os resultados de convergência para passos preditores-corretores usaremos alguns lemas técnicos. O lema seguinte, de Ye [40], é uma variação mais estrita de um lema originalmente apresentado por Monteiro e Adler [24].

Lema 2: *Sejam p e q dois vetores reais de dimensão n tais que $p^T q \geq 0$. Então*

$$\|pq\| \leq \frac{\|p+q\|^2}{\sqrt{8}}.$$

Demonstração:

$$\begin{aligned} \|pq\|^2 &= \sum_{i=1}^n (p_i q_i)^2 \\ &\leq \left(\sum_{p_i q_i > 0} p_i q_i \right)^2 + \left(\sum_{p_i q_i < 0} p_i q_i \right)^2 \\ &\leq 2 \left(\sum_{p_i q_i > 0} p_i q_i \right)^2 \\ &\leq 2 \left(\sum_{p_i q_i > 0} (p_i + q_i)^2 / 4 \right)^2 \\ &\leq \|p+q\|^4 / 8. \end{aligned}$$

Lema 3: *Suponha que $\|xz - \mu e\| \leq \theta \mu$ e que $\hat{\mu} = (1 - \tau) \mu$. Então*

$$\|\delta x \delta z\| \leq \frac{(\theta + \sqrt{n} \tau)^2}{\sqrt{8} (1 - \theta)} \mu.$$

Demonstração: Defina $D = Z^{1/2} X^{-1/2}$, $p = D \delta x$, $q = D^{-1} \delta z$. Multiplique a última equação do sistema (2.3) por $(XZ)^{-1/2}$ de modo que $p + q = D \delta x + D^{-1} \delta z = r = (xz)^{-1/2} (\hat{\mu} e - xz)$. Observando que $\|xz - \mu e\| \leq \theta \mu$ implica que $x_i z_i \geq (1 - \theta) \mu$ e

usando o Lema 2: temos que

$$\begin{aligned}
\|\delta x \delta z\| &\leq \|\delta x + \delta z\|^2 / \sqrt{8} = \left\| (xz)^{-1/2} (\hat{\mu}e - xz) \right\|^2 / \sqrt{8} \\
&= \frac{1}{\sqrt{8}} \sum_{i=1}^n \frac{(x_i z_i - \hat{\mu})^2}{x_i z_i} \\
&\leq \frac{1}{\sqrt{8}} \sum_{i=1}^n \frac{(x_i z_i - \hat{\mu})^2}{(1-\theta)\mu} \\
&= \frac{1}{\sqrt{8}(1-\theta)\mu} \|xz - \hat{\mu}e\|^2 \\
&= \frac{1}{\sqrt{8}(1-\theta)\mu} \|(xz - \mu e) + (\mu - \hat{\mu})e\|^2 \\
&\leq \frac{1}{\sqrt{8}(1-\theta)\mu} (\|xz - \mu e\| + \|(\mu - \hat{\mu})e\|)^2 \\
&\leq \frac{1}{\sqrt{8}(1-\theta)\mu} (\theta\mu + \sqrt{n}\tau\mu)^2 \\
&= \frac{(\theta + \sqrt{n}\tau)^2}{\sqrt{8}(1-\theta)} \mu.
\end{aligned}$$

Lema 4: *Suponha que $\|xz - \mu e\| \leq \theta\mu$ e que $\hat{\mu} = (1 - \tau)\mu$. Então*

$$\|\Delta x \Delta z\| \leq \frac{1}{\sqrt{8}(1-\theta)} \left(\theta + \sqrt{n}\tau + \frac{(\theta + \sqrt{n}\tau)^2}{\sqrt{8}(1-\theta)} \right)^2 \mu.$$

Demonstraçãõ: Novamente defina $D = Z^{1/2}X^{-1/2}$, $p = D\Delta x$, $q = D^{-1}\Delta z$. Multiplique a última equaçãõ do sistema (3.1) por $(xz)^{-1/2}$ de modo que $p+q = D\Delta x + D^{-1}\Delta z = r = (xz)^{-1/2}(\hat{\mu}e - xz - \delta x \delta z)$. Observe que $-\theta\mu \leq x_i z_i - \mu \leq \theta\mu$ e assim pelo Lema

2: e Lema 3: temos que

$$\begin{aligned}
\|\Delta x \Delta z\| &\leq \|\Delta x + \Delta z\|^2 / \sqrt{8} = \left\| (xz)^{-1/2} (\hat{\mu}e - xz - \delta x \delta z) \right\|^2 / \sqrt{8} \\
&= \frac{1}{\sqrt{8}} \sum_{i=1}^n \frac{(x_i z_i - \hat{\mu} + \delta x_i \delta z_i)^2}{x_i z_i} \\
&\leq \frac{1}{\sqrt{8}} \sum_{i=1}^n \frac{(x_i z_i - \hat{\mu} + \delta x_i \delta z_i)^2}{(1-\theta)\mu} \\
&= \frac{1}{\sqrt{8}(1-\theta)\mu} \|xz - \hat{\mu}e + \delta x \delta z\|^2 \\
&= \frac{1}{\sqrt{8}(1-\theta)\mu} \|(xz - \mu e) + (\mu - \hat{\mu})e + \delta x \delta z\|^2 \\
&\leq \frac{1}{\sqrt{8}(1-\theta)\mu} (\|xz - \mu e\| + \|(\mu - \hat{\mu})e\| + \|\delta x \delta z\|)^2 \\
&\leq \frac{1}{\sqrt{8}(1-\theta)\mu} \left(\theta\mu + \sqrt{n}\tau\mu + \frac{(\theta + \sqrt{n}\tau)^2}{\sqrt{8}(1-\theta)}\mu \right)^2 \\
&= \frac{1}{\sqrt{8}(1-\theta)} \left(\theta + \sqrt{n}\tau + \frac{(\theta + \sqrt{n}\tau)^2}{\sqrt{8}(1-\theta)} \right)^2 \mu.
\end{aligned}$$

Teorema 3: *Sejam τ e θ reais tais que*

$$\frac{1}{\sqrt{8}(1-\theta)} \left(\theta + \sqrt{n}\tau + \frac{(\theta + \sqrt{n}\tau)^2}{\sqrt{8}(1-\theta)} \right)^2 + \frac{(\theta + \sqrt{n}\tau)^2}{\sqrt{8}(1-\theta)} \leq \theta(1-\tau)$$

Se $\|xz - \mu e\| \leq \theta\mu$ e $\hat{\mu} = (1-\tau)\mu$ então $\|\hat{x}\hat{z} - \hat{\mu}e\| \leq \theta\hat{\mu}$.

Demonstração:

$$\begin{aligned}
\hat{x}_i \hat{z}_i - \hat{\mu} &= (x_i + \Delta x_i)(z_i + \Delta z_i) - \hat{\mu} \\
&= x_i z_i + x_i \Delta z_i + z_i \Delta x_i + \Delta x_i \Delta z_i - \hat{\mu} \\
&= \Delta x_i \Delta z_i - \delta x_i \delta z_i
\end{aligned}$$

e, portanto,

$$\begin{aligned}
\|\hat{x}\hat{z} - \hat{\mu}e\| &= \|\Delta x \Delta z - \delta x \delta z\| \leq \|\Delta x \Delta z\| + \|\delta x \delta z\| \\
&\leq \left(\frac{1}{\sqrt{8}(1-\theta)} \left(\theta + \sqrt{n}\tau + \frac{(\theta + \sqrt{n}\tau)^2}{\sqrt{8}(1-\theta)} \right)^2 + \frac{(\theta + \sqrt{n}\tau)^2}{\sqrt{8}(1-\theta)} \right) \mu \\
&\leq \theta(1-\tau)\mu = \theta\hat{\mu}.
\end{aligned}$$

Observe que se $\theta = \frac{1}{4}$ e $\tau = \min(\frac{1}{5\sqrt{n}}, \frac{1}{5\sqrt{13}})$ então a condição do Teorema 3: é satisfeita.

Isso mostra que o Passo 3 do Algoritmo do Capítulo 5 está bem definido se a vizinhança usada é $\mathcal{N}_2(\frac{1}{4})$. A escolha $\tau = \frac{1}{5\sqrt{n}}$ garante que

$$\frac{g_{k+1}}{g_k} \leq \left(1 - \frac{1}{5\sqrt{n}}\right),$$

o que por sua vez garante que o algoritmo para em $\mathcal{O}(\sqrt{n}L)$ passos.

Quando a vizinhança \mathcal{N}_2 é usada, a minimização subjacente no Passo 3 é

$$\begin{aligned}
&\min \hat{g} = \alpha n \mu + (1 - \alpha) g \\
\text{s.a.} \quad &\hat{w} \in \mathcal{N}_2\left(\frac{1}{4}\right)
\end{aligned}$$

mas, nesse caso,

$$\begin{aligned}
\hat{w} \in \mathcal{N}_2\left(\frac{1}{4}\right) &\Leftrightarrow \\
\|\hat{x}\hat{z} - \hat{\mu}e\|^2 &\leq \frac{9}{16}\hat{\mu}^2 \Leftrightarrow \\
\sum_{i=1}^n (\hat{x}_i \hat{z}_i - \hat{\mu})^2 &\leq \frac{9}{16}\hat{\mu}^2 \Leftrightarrow \\
\sum_{i=1}^n ((x_i + \alpha \Delta x_i)(z_i + \alpha \Delta z_i) - \hat{\mu})^2 &\leq \frac{9}{16}\hat{\mu}^2 \Leftrightarrow \\
\sum_{i=1}^n ((x_i + \alpha(\psi_2^{x_i} \hat{\mu}^2 + \psi_1^{x_i} \hat{\mu} + \psi_0^{x_i}))(z_i + \alpha(\psi_2^{z_i} \hat{\mu}^2 + \psi_1^{z_i} \hat{\mu} + \psi_0^{z_i})) - \hat{\mu})^2 - \frac{9}{16}\hat{\mu}^2 &\leq 0.
\end{aligned}$$

Essa última inequação pode ser reagrupada de modo a formar um polinômio de grau 8 nas variáveis α e $\hat{\mu}$, porque os vetores ψ não dependem de $\hat{\mu}$ e α e foram previamente calculados. Para cada $\alpha \in (0, 1]$ fixado, o problema de otimização acima pode ser resolvido com precisão pois a solução $\hat{\mu}$ é a menor raiz positiva de um polinômio. Uma vez que é computacionalmente barato encontrar raízes de um polinômio para cada α , podemos resolver esse subproblema com bastante precisão — e fizemos isso em nossas implementações descritas no Capítulo 8.

Generalizaremos agora o resultado anterior para a vizinhança mais larga \mathcal{N}_{∞}^- .

Lema 5: *Sejam p e q dois vetores reais de dimensão n tais que $p^T q = 0$. Então*

$$-\frac{\|p+q\|^2}{4} \leq p_i q_i \leq \frac{(p_i+q_i)^2}{4}$$

Demonstração: Reproduziremos a demonstração de Ye [40]. A inequação da direita segue-se de $p_i + q_i = r_i$. Para a inequação da esquerda

$$\begin{aligned} p_i q_i &\geq \sum_{p_i q_i < 0} p_i q_i \\ &= - \sum_{p_i q_i > 0} p_i q_i \quad (\text{já que } p^T q = 0) \\ &\geq - \sum_{p_i q_i > 0} \frac{r_i^2}{4} \quad (\text{usando a inequação da direita}) \\ &\geq - \|r\|^2 / 4 \end{aligned}$$

Teorema 4: *Sejam τ e θ reais tais que*

$$-\frac{\left(\theta + \tau\sqrt{n} + \frac{(\theta+\tau)^2}{4\theta}\right)^2}{\theta} + \frac{(\theta + \tau)^2}{4\theta} + (1 - \tau) \geq \theta(1 - \tau).$$

Se $xz \geq \theta\mu$ e $\hat{\mu} = (1 - \tau)\mu$ então $\hat{x}\hat{z} \geq \theta\hat{\mu}$.

Demonstraçãõ: De acordo com lema anterior temos que

$$\begin{aligned}
\delta x_i \delta z_i &\leq \frac{(x_i z_i - \hat{\mu})^2}{4(x_i z_i)} \leq \frac{(x_i z_i - \hat{\mu})^2}{4\theta\mu} \\
&= \frac{((x_i z_i - \mu) + (\mu - \hat{\mu}))^2}{4\theta\mu} \\
&\leq \frac{(\theta\mu + \tau\mu)^2}{4\theta\mu} \\
&= \frac{(\theta + \tau)^2}{4\theta} \mu
\end{aligned}$$

e

$$\begin{aligned}
\Delta x_i \Delta z_i &\geq - \frac{\| (xz)^{-1/2} (\hat{\mu}e - xz - \delta x \delta z) \|^2}{4} \\
&= - \sum_{i=1}^n \frac{(\hat{\mu} - x_i z_i - \delta x_i \delta z_i)^2}{x_i z_i} \\
&\geq - \sum_{i=1}^n \frac{(\hat{\mu} - x_i z_i - \delta x_i \delta z_i)^2}{\theta\mu} \\
&= - \frac{\| (\mu e - xz) + (\hat{\mu} - \mu) e - \delta x \delta z \|^2}{\theta\mu} \\
&\geq - \frac{(\| \mu e - xz \| + \| \tau \mu e \| + \| \delta x \delta z \|)^2}{\theta\mu} \\
&\geq - \frac{\left(\theta\mu + \tau\sqrt{n}\mu + \frac{(\theta+\tau)^2}{4\theta}\mu \right)^2}{\theta\mu} \\
&= - \frac{\left(\theta + \tau\sqrt{n} + \frac{(\theta+\tau)^2}{4\theta} \right)^2}{\theta\mu} \mu.
\end{aligned}$$

Assim,

$$\begin{aligned}
\hat{x}_i \hat{z}_i &= \Delta x_i \Delta z_i - \delta x_i \delta z_i + \hat{\mu} \\
&\geq \left(-\frac{\left(\theta + \tau\sqrt{n} + \frac{(\theta+\tau)^2}{4\theta}\right)^2}{\theta} + \frac{(\theta + \tau)^2}{4\theta} + (1 - \tau) \right) \mu \\
&\geq \theta(1 - \tau) \mu \\
&= \theta \hat{\mu}.
\end{aligned}$$

Observe que uma escolha possível que satisfaz a condição no Teorema 4: é

$$\begin{aligned}
\theta &= \frac{1}{5}, \\
\tau &= \min\left(\frac{1}{2n}, \frac{1}{30}\right).
\end{aligned}$$

Novamente esse teorema provou que o Passo 3 no algoritmo geral está bem definido se a vizinhança usada for $\mathcal{N}_\infty^-\left(\frac{1}{5}\right)$. A escolha $\tau = 1/(2n)$ garante que

$$\frac{g_{k+1}}{g_k} \leq \left(1 - \frac{1}{2n}\right),$$

o que por sua vez garante que o algoritmo termina em $\mathcal{O}(nL)$ passos.

Quando usamos a vizinhança \mathcal{N}_∞^- , a minimização subjacente no Passo 3 é

$$\begin{aligned}
\min \hat{g} &= \alpha n \mu + (1 - \alpha) g \\
\text{s.a. } \hat{w} &\in \mathcal{N}_\infty^-(0.2)
\end{aligned}$$

e de modo similar a condição de que $\hat{w} \in \mathcal{N}_\infty^-(0.2)$ é equivalente a n inequações

$$\hat{x}_i \hat{z}_i \geq 0.2 \hat{\mu}.$$

Se usarmos uma parametrização de segunda ordem, cada uma dessas inequações é equi-

valente a uma inequação polinomial de quarta ordem

$$(x_i + \alpha (\psi_2^{x_i} \hat{\mu}^2 + \psi_1^{x_i} \hat{\mu} + \psi_0^{x_i})) (z_i + \alpha (\psi_2^{z_i} \hat{\mu}^2 + \psi_1^{z_i} \hat{\mu} + \psi_0^{z_i})) - 0.2\hat{\mu} \geq 0.$$

A minimização nesse caso é muito mais difícil computacionalmente. Fixando $\alpha \in (0, 1]$, cada inequação tem até dois intervalos como conjunto solução, associado às raízes do polinômio, e o conjunto solução geral é obtido por cálculo de intervalos usando até $2n$ intervalos, em um cálculo de $\mathcal{O}(n)$ operações aritméticas para cada α fixo. O $\hat{\mu}$ ótimo associado fica então disponível como o menor elemento positivo do conjunto solução geral — de modo que temos que limitar o número de α diferentes avaliados para obter uma minimização aproximada. O algoritmo resultante para a vizinhança $\mathcal{N}_{\infty}^-(0.2)$ foi implementado e os resultados computacionais estão mostrados no Capítulo 8.

Capítulo 7

Contexto auto-dual

Apresentamos agora a extensão para um contexto auto-dual que foi de fato usada para as implementações, para dar conta da questão do ponto inicial factível. A fim de poder lidar com variáveis canalizadas decidimos separar explicitamente as variáveis canalizadas das não-canalizadas. Essa é a praxe em implementações práticas e com isso a tradução dos resultados algébricos pode ser feita diretamente para a implementação.

Definimos o problema primal

$$(LP) \quad \begin{array}{l} \min c_1^T v + c_2^T x \\ \text{s.a.} \left\{ \begin{array}{l} A_1 v + A_2 x = b \\ 0 \leq v \leq u \\ x \geq 0 \end{array} \right. \end{array}$$

onde $c_1, v, u \in \mathbb{R}^k$, $c_2, x \in \mathbb{R}^{n-k}$, $b \in \mathbb{R}^m$, $A_1 \in \mathbb{R}^{m \times k}$, $A_2 \in \mathbb{R}^{m \times (n-k)}$ e definimos o correspondente problema dual

$$(LD) \quad \begin{array}{l} \max b^T y - u^T w \\ \text{s.a.} \left\{ \begin{array}{l} A_1^T y - w \leq c_1 \\ A_2^T y \leq c_2 \\ w \geq 0 \end{array} \right. \end{array}$$

onde $y \in \mathbb{R}^m$, $w \in \mathbb{R}^k$, e definimos o problema auto-dual homogêneo associado

$$\begin{array}{l}
 \text{(HLP)} \quad \min c_0 \theta \\
 \text{s.a.} \quad \left\{ \begin{array}{l}
 (1) \quad A_1 v + A_2 x - b \tau - r_1 \theta = 0 \\
 (2) \quad -v + u \tau - r_2 \theta \geq 0 \\
 (3) \quad -A_1^T y + w + c_1 \tau - r_3 \theta \geq 0 \\
 (4) \quad -A_2^T y + c_2 \tau - r_4 \theta \geq 0 \\
 (5) \quad b^T y - u^T w - c_1^T v - c_2^T x - r_5 \theta \geq 0 \\
 (6) \quad r_1^T y + r_2^T w + r_3^T v + r_4^T x + r_5 \tau = -c_0 \\
 \quad \quad \quad w, \quad v, \quad x, \quad \tau, \quad \geq 0
 \end{array} \right.
 \end{array}$$

onde, designando por s , h , z e κ as variáveis de folga das inequações (2), (3), (4) e (5) respectivamente,

$$\begin{aligned}
 r_1 &= (A_1 v_0 + A_2 x_0 - b \tau_0) / \theta_0 \\
 r_2 &= (u_0 \tau_0 - v_0 - s_0) / \theta_0 \\
 r_3 &= (c_1 \tau_0 - A_1^T y_0 - h_0 + w_0) / \theta_0 \\
 r_4 &= (c_2 \tau_0 - A_2^T y_0 - z_0) / \theta_0 \\
 r_5 &= (b^T y_0 - u_0^T w_0 - c_1^T v_0 - c_2^T x_0 - \kappa_0) / \theta_0 \\
 c_0 &= (v_0^T h_0 + x_0^T z_0 + s_0^T w_0 + \tau_0 \kappa_0) / \theta_0
 \end{aligned}$$

e onde $v_0 > 0$, $x_0 > 0$, $s_0 > 0$, $z_0 > 0$, $w_0 > 0$ e y_0 são pontos iniciais quaisquer com $\tau_0 = \kappa_0 = \theta_0 = 1$, de modo que $(y_0, w_0, v_0, x_0, \tau_0, \theta_0, s_0, h_0, z_0, \kappa_0)$ satisfaz (HLP). Definimos também \mathcal{F}_h^0 como o conjunto de todos os pontos $(y_0, w_0, v_0, x_0, \tau_0, \theta_0, s_0, h_0, z_0, \kappa_0)$ que satisfazem (HLP) tais que

$$(w, v, x, \tau, s, h, z, \kappa) > 0.$$

Desse modo, a trajetória (auto-) central para (HLP) é definida como

$$\mathcal{C} = \left\{ (y, w, v, x, \tau, \theta, s, h, z, \kappa) \in \mathcal{F}_h^0 : \begin{pmatrix} vh \\ xz \\ ws \\ \tau\kappa \end{pmatrix} = \mu e \right\}$$

e definimos a vizinhança euclidiana da trajetória central como

$$\mathcal{N}_2(\beta) = \left\{ (y, w, v, x, \tau, \theta, s, h, z, \kappa) \in \mathcal{F}_h^0 : \left\| \begin{pmatrix} vh \\ xz \\ ws \\ \tau\kappa \end{pmatrix} - \mu e \right\| \leq \beta \mu \right\}$$

A extensão para outras vizinhanças pode ser feita de um modo similar ao que fizemos no contexto primal-dual.

Utilizando exatamente os mesmos argumentos de [39], mas agora no contexto de variáveis canalizadas, pode-se provar que (HLP):

- é auto-dual no sentido de que é um problema linear cujo dual é idêntico a (HLP);
- tem um ponto interior factível $(y_0, w_0, v_0, x_0, \tau_0, \theta_0, s_0, h_0, z_0, \kappa_0)$ que está exatamente na trajetória central, para

$$\mu_0 = \frac{v_0^T h_0 + x_0^T z_0 + w_0^T s_0 + \tau_0 \kappa_0}{n + k + 1};$$

- tem uma solução ótima e o conjunto solução é limitado;
- tem uma solução estritamente auto-complementar, i.e., uma solução

$$(y^*, w^*, v^*, x^*, \tau^*, \theta^* = 0, s^*, h^*, z^*, \kappa^*)$$

tal que

$$\begin{pmatrix} v^* + h^* \\ x^* + z^* \\ w^* + s^* \\ \tau^* + \kappa^* \end{pmatrix} > 0$$

Além disso,

- (LP) tem uma solução factível e limitada se e somente se $\tau^* > 0$. Nesse caso, $(v^*/\tau^*, x^*/\tau^*)$ é uma solução ótima para (LP) e $(y^*/\tau^*, w^*/\tau^*)$ é uma solução ótima para (LD).
- Se $\tau^* = 0$ então:
 - a) se $c_1^T v^* + c_2^T x^* < 0$ então (LP) é infactível;
 - b) se $b^T y^* - u^T w^* < 0$ então (LD) é infactível.(Os casos (a) e (b) não são mutuamente exclusivos).

Todas essas características implicam que quando trabalhamos em um contexto auto-dual podemos sempre encontrar uma solução finita para (HLP), com a qual podemos resolver completamente o problema original.

Há uma consequência muito prática do fato de (HLP) ser auto-dual. Analisando as condições de otimalidade para (HLP) podemos ver que elas são quase idênticas às do problema original (LP), no sentido de que temos que considerar apenas a forma primal de (HLP), especialmente no que diz respeito a todas as rotinas de álgebra linear.

Como consequência imediata todos os resultados dos capítulos anteriores permanecem válidos, mas as parametrizações de primeira e segunda ordem têm que ser refeitas nesse contexto auto-dual, a fim de permitir uma implementação adequada. Nesse caso as fórmulas e notações tornam-se muito carregadas, de modo que essas parametrizações foram colocadas no apêndice A, e apenas para algumas vizinhanças.

Pode parecer que o sistema envolvido em (HLP) é muito grande. No entanto, o sistema que fornece a direção de Newton associada à condição de Karush-Kuhn-Tucker para (HLP) é

$$A_1 d_v + A_2 d_x - b d_\tau - r_1 d_\theta = 0 \quad (1)$$

$$-d_v + u d_\tau - r_2 d_\theta - d_s = 0 \quad (2)$$

$$-A_1^T d_y + d_w + c_1 d_\tau - r_3 d_\theta - d_h = 0 \quad (3)$$

$$-A_2^T d_y + c_2 d_\tau - r_4 d_\theta - d_z = 0 \quad (4)$$

$$b^T d_y - u^T d_w - c_1^T d_v - c_2^T d_x - r_5 d_\theta - d_\kappa = 0 \quad (5)$$

$$r_1^T d_y + r_2^T d_w + r_3^T d_v + r_4^T d_x + r_5 d_\tau = 0 \quad (6)$$

$$S d_w + W d_s = \mu e_1 - w s \quad (7)$$

$$H d_v + V d_h = \mu e_2 - v h \quad (8)$$

$$Z d_x + X d_z = \mu e_3 - x z \quad (9)$$

$$\kappa d_\tau + \tau d_\kappa = \mu - \tau \kappa \quad (10)$$

e podemos, após alguma manipulação algébrica, mostrar que esse sistema pode ser resolvido usando a solução de um sistema menor similar a $AA^T x = \vec{b}$ onde $A = [A_1 : A_2]$, e assim o esforço computacional é da mesma ordem de grandeza dos algoritmos primal-dual usuais. Essa manipulação algébrica está explicitada no apêndice A.

Capítulo 8

Resultados computacionais

Conforme mencionamos na introdução, estamos interessados em fazer comparações justas entre implementações práticas. Para isso, tomamos como base a implementação PCx [4] do algoritmo preditor-corretor de Mehrotra [22], para a qual o código em C está disponível. Mas fizemos algumas pequenas mudanças. Primeiro, a heurística de Mehrotra para o ponto inicial não gera um ponto inicial centrado, mas mexendo minimamente nos parâmetros de sua heurística pudemos gerar pontos iniciais que são auto-centrados com relação a \mathcal{N}_2 . Em segundo lugar, o PCx permite correções de ordem superior de Gondzio [11] — nós inibimos essa característica no arquivo de configuração mas permitimos o refinamento por gradiente conjugado presente na implementação original. Denominamos essa implementação de referência PCx-r.

Em seguida introduzimos no código original, desviando do algoritmo principal, o código para nossos algoritmos, desse modo herdando as mesmas rotinas para pré-processamento (leitura de dados, condicionamento por escalonamento, reordenação de linhas e colunas, tratamento de colunas densas e fatoração simbólica de Cholesky), ponto inicial, todas as rotinas de álgebra linear (rotinas de Ng e Peyton [26] para resolução de sistemas lineares esparsos, bem como refinamento por gradiente conjugado), critérios de parada e saída dos dados.

Na fase de desenvolvimento desse trabalho fizemos várias implementações, para di-

versas estratégias de escolha das vizinhanças, incluindo tipo de vizinhança e parâmetro β , com passo estritamente preditor ou passo preditor-corretor, permitindo ou não que o tamanho de passo α fosse otimizado e usando estratégias diferentes para a resolução aproximada do subproblema não-linear de otimização. Os resultados computacionais de todas essas implementações é, portanto, bastante extenso e sua apresentação em tabelas talvez trouxesse mais confusão do que esclarecimento. Optamos, em vez disso, por escolher três implementações que ilustram melhor algumas conclusões possíveis.

As implementações excluídas o foram pelas seguintes razões:

- Não apresentamos passos estritamente preditores. Observamos que essa família de implementações mais simples apresenta, como já é sabido, um número de iterações e de tempo computacional que é maior do que seus correspondentes algoritmos preditores-corretores.
- As vizinhanças de Jansen não apresentaram um desempenho diferente de seus correspondentes na vizinhança euclidiana, em número de iterações ou tempo de CPU.

Todos os códigos foram compilados com as mesmas opções de compilação e no mesmo computador. Com essa estratégia, quaisquer diferenças no número de iterações e no tempo de CPU podem ser atribuídas apenas às implementações dos algoritmos. Além disso, dessa forma os tempos de CPU não são importantes em termos absolutos, mas apenas suas relações de uma implementação para outra.

A cada iteração de nossos algoritmos fizemos uma transformação que toma um iterando no ambiente *primal-dual* do PCx e o transforma em um iterando *auto-dual*, a partir do qual computamos o próximo iterando e então transformamos de volta para o ambiente primal-dual. Essa transformação é computacionalmente barata, já que envolve $\mathcal{O}(n)$ operações aritméticas por iteração.

É importante ressaltar que todas as implementações exigem rotinas eficientes para se encontrar todas as raízes de um polinômio. Como isso não envolve o tamanho do problema n nem dos dados L , essas rotinas não afetam a complexidade do algoritmo.

Além disso, para polinômios de grau menor que cinco foi usada uma rotina mista, que combina resolução por radicais, de modo robusto, com refinamento iterativo.

Fizemos três implementações, onde os valores do parâmetro β foram escolhidos com base nas demonstrações de convergência.

- A primeira, que chamamos de PCx-N2a, usa $\mathcal{N}_2(0.25)$ como vizinhança da trajetória central e fixa $\alpha = 1$, otimizando apenas em μ . Com isso no Passo 3 do algoritmo a função objetivo e a restrição a que o próximo iterando deve satisfazer passam a ser

$$\text{s.a.} \left\{ \begin{array}{l} \min_{\alpha, \mu} n\mu \\ \sum_{i=1}^n (\hat{x}_i \hat{z}_i - \mu)^2 \leq \frac{9}{16} \mu^2 \\ \text{ou} \\ \sum_{i=1}^n ((x_i + \psi_{2,i}^x \mu^2 + \psi_{1,i}^x \mu + \psi_{0,i}^x) (z_i + \psi_{2,i}^z \mu^2 + \psi_{1,i}^z \mu + \psi_{0,i}^z) - \mu)^2 \leq \frac{9}{16} \mu^2, \end{array} \right.$$

na notação do contexto primal-dual (no contexto auto-dual, que foi efetivamente usado, essa notação é mais carregada e está explicitada na seção A.2).

- A segunda, PCx-N2b permite que tanto α quanto μ sejam otimizados, ou seja, a otimização agora é

$$\text{s.a.} \left\{ \begin{array}{l} \min_{\alpha, \mu} \alpha n\mu + (1 - \alpha) g \\ \sum_{i=1}^n (\hat{x}_i \hat{z}_i - \mu)^2 \leq \frac{9}{16} \mu^2 \\ \text{ou} \\ \sum_{i=1}^n ((x_i + \alpha (\psi_{2,i}^x \mu^2 + \psi_{1,i}^x \mu + \psi_{0,i}^x)) (z_i + \alpha (\psi_{2,i}^z \mu^2 + \psi_{1,i}^z \mu + \psi_{0,i}^z)) - \mu)^2 \leq \frac{9}{16} \mu^2. \end{array} \right.$$

A otimização em α e μ foi feita discretizando α entre 0 e 1.2 e para cada valor de α fizemos a otimização em μ como variável real.

- A terceira, PCx-SN usa $\mathcal{N}_\infty^-(0.2)$ e também fixa $\alpha = 1$. Nesse caso a otimização é

$$\begin{aligned} & \min_{\alpha, \mu} n\mu \\ \text{s.a. } & \left\{ \begin{array}{l} (x_i + \psi_{2,i}^x \mu^2 + \psi_{1,i}^x \mu + \psi_{0,i}^x) (z_i + \psi_{2,i}^z \mu^2 + \psi_{1,i}^z \mu + \psi_{0,i}^z) \leq 0.2\mu \\ \text{para } i = 1, \dots, n. \end{array} \right. \end{aligned}$$

Essa otimização envolve um conjunto de n restrições em μ ; cada restrição tem por solução um conjunto formado por até dois intervalos reais a que μ deve pertencer, e com isso o conjunto factível global é a interseção de n conjuntos de até dois intervalos. Computacionalmente a geração desse conjunto factível global é relativamente complexa e cara, no sentido que, a cada iteração, envolve n vezes a obtenção de todas as raízes reais de um polinômio de grau quatro.

Observe que os resultados de convergência foram previamente demonstrados para todos esses algoritmos.

Alguns detalhes adicionais: os problemas que estudamos formam um subconjunto da Netlib, de médios a grandes, alguns com variáveis canalizadas e todos factíveis. Decidimos não estudar problemas infactíveis porque desejávamos usar o mesmo critério de parada do PCx. Na verdade, devido ao contexto auto-dual, poderíamos ter usado um critério de parada que tem se mostrado mais eficiente para problemas infactíveis [37].

A tabela a seguir mostra o número de iterações e os tempos de CPU em segundos para cada implementação. A coluna L indica o número de elementos não nulos no fator de Cholesky L. Todos os testes foram feitos usando um processador Pentium III a 500 MHz sob Windows NT e usando um compilador C Borland 5.0.

				PCx-r		PCx-SN		PCx-N2a		PCx-N2b	
Problema	Lins	Cols	L	iter.	CPU	iter.	CPU	iter.	CPU	iter.	CPU
25FV47	821	1571	33809	30	1.83	24	3.19	44	4.10	44	4.22
BANDM	305	472	3936	19	0.14	16	0.33	30	0.42	30	0.49
BNL2	2324	3489	81275	40	8.94	34	13.32	67	19.19	67	19.33
BOEING1	351	384	5725	21	0.25	29	1.11	54	1.26	54	1.39
BOEING2	166	143	1912	18	0.03	18	0.24	34	0.26	34	0.32
BORE3D	233	315	1034	18	0.03	15	0.11	28	0.09	28	0.17
CAPRI	271	353	3962	24	0.17	20	2.65	42	0.56	42	0.64
CYCLE	1903	2857	56102	45	4.19	22	16.72	51	8.47	51	8.6
CZPROB	929	3523	3520	32	0.58	21	13.77	41	2.18	41	2.27
FIT2P	3000	13525	3000	21	4.57	22	125.72	59	47.31	59	47.61
FFFFFF800	524	854	9573	33	0.55	26	5.95	51	1.56	51	1.69
FORPLAN	161	421	3304	28	0.25	23	3.06	47	0.76	47	0.89
PILOT	1441	3652	200812	43	32.56	62	130.16	135	122.88	135	123.24
PILOT.WE	722	2789	15605	45	1.70	38	30.30	79	6.77	79	7.00

Como tínhamos mencionado antes, na implementação PCx-SN conseguimos obter menos iterações na maior parte dos problemas, apesar de os tempos de CPU serem maiores. Há dois motivos para isso. Primeiro e mais importante, a cada iteração temos que resolver sete sistemas lineares usando o mesmo fator de Cholesky a fim de calcular a parametrização preditor-corretor, contra duas resoluções de sistema no algoritmo preditor-corretor usual. Em segundo lugar, nessa primeira versão de nossa implementação ainda há partes importantes do código que precisam ser reescritas para aumentar a velocidade, tais como as rotinas de cálculos de intervalos e as rotinas que acham raízes de polinômios. Isso é corroborado pela diferença em tempo de CPU entre PCx-N2a e PCx-SN — o primeiro não usa cálculo de intervalos e tem tempo de CPU menor, apesar de fazer mais iterações.

É interessante observar que PCx-N2a e PCx-N2b fizeram exatamente o mesmo número de iterações. Isso ocorreu porque apesar da otimização ter sido permitida em PCx-N2b

também para o tamanho de passo α , o ótimo sempre foi alcançado para $\alpha = 1$. Nossa interpretação é a de que isso ocorre por se tratar de algoritmos de pontos factíveis — para algoritmos de pontos infactíveis de redução de potencial, por exemplo, isso não ocorre.

De modo similar com PCx-SN, fizemos também uma outra implementação permitindo que o tamanho de passo α fosse incluído na otimização, mas da mesma forma o ótimo foi alcançado para $\alpha = 1$, e por isso não a apresentamos aqui.

Capítulo 9

Conclusões

Conforme tínhamos proposto na Seção 1.2, com a parametrização do próximo iterando em termos de α e μ conseguimos estabelecer um ambiente analítico no qual as escolhas desses parâmetros pode ser enunciada como um claro problema de otimização — o que dispensa o uso de heurísticas para essas escolhas. Isso não significa que as heurísticas usadas por Mehrotra não sejam boas — mas permanece a necessidade de substituir essas heurísticas por ferramentas analíticas.

Mais do que simplesmente introduzir um recurso analítico, mostramos que o tipo de parametrização que fizemos é útil como ferramenta, na medida em que permite avaliar e melhorar as demais partes de um dado algoritmo, já que agora elas podem ser expressas em termos dessa parametrização. Isso ainda não tinha sido feito explicitamente antes, e essa talvez seja a contribuição mais original desse trabalho.

Podemos concluir também que o uso de um contexto auto-dual, apesar de permitir um tratamento adequado da questão do ponto inicial e da detecção da infactibilidade, apresenta a grande desvantagem de exigir um número maior de resoluções de sistemas a cada iteração, e esse preço na prática é proibitivo, já que o contexto auto-dual por si só não conseguiu, em nossos experimentos, reduzir o número de iterações a ponto de compensar o esforço adicional de resoluções de sistema. Esse esforço extra de resolver sistemas adicionais não compensa porque para os problemas da Netlib os tempos de

CPU necessários para cada fatoração de Cholesky *não* são muito maiores do que os tempos necessários para se resolver cada sistema positivo-definido, já dispondo do fator de Cholesky — e a razão disso é que a maioria dos problemas da Netlib é esparsa.

Também concluímos que a otimização envolvendo vizinhanças largas deve ser feita com extremo cuidado, tendo em vista sua complexidade computacional maior — mas a pesquisa nessa direção merece continuar, já que a redução no número de iterações é significativa.

Há dois desenvolvimentos possíveis usando a idéia de parametrizar o próximo iterando do modo que fizemos.

Um consiste em fazer com que a parametrização para o algoritmo preditor-corretor use $\mu = 0$ no passo preditor e calcular μ apenas no passo corretor. Isso forçaria a parametrização a ser de primeira ordem em μ — e portanto mais barata computacionalmente por exigir menos resoluções de sistema — e a faria mais similar ao algoritmo de Mehrotra.

Outro consiste em trabalhar fora do contexto auto-dual, diretamente com um algoritmo de pontos inactíveis, a fim de evitar a resolução extra de sistemas. Nesse contexto seria possível também usar algoritmos de redução de potencial em vez de trajetória central.

Apêndice A

Parametrização no contexto auto-dual

Neste capítulo desenvolveremos explicitamente a parametrização dos passos em cada iteração para o contexto auto-dual, com dois objetivos. Primeiro, queremos um desenvolvimento que possa ser traduzido facilmente para um programa escrito em C, de modo que faremos substituições que não só tornam a notação mais simples mas que também correspondem a etapas intermediárias que teriam que ser feitas de qualquer forma durante a implementação. Segundo, mostraremos que apesar de o sistema envolvido em (HLP) ser grande, sua resolução é equivalente a de um sistema menor, de mesmo tamanho dos algoritmos primal-dual.

Cabe observar que a maior parte do desenvolvimento algébrico que se segue, por ser muito complexo e com isso criar muitas oportunidades de erro humano, foi feita usando um pacote de processamento simbólico que faz parte do Maple acoplado ao processador de texto Scientific Workplace 3.0. As fórmulas resultantes, depois de verificadas, foram copiadas diretamente para as rotinas em C.

A.1 Parametrização de primeira ordem

No contexto auto-dual com variáveis canalizadas, dado um ponto factível $(y, w, v, x, \tau, \theta, s, h, \kappa)$, o sistema linear em $\delta_y, \delta_w, \delta_v, \delta_x, \delta_\tau, \delta_\theta, \delta_s, \delta_h, \delta_z$ e δ_κ que precisamos resolver para obter as direções de Newton do sistema definido pelas condições de Karush-Kuhn-Tucker, é dado por

$$A_1\delta_v + A_2\delta_x - b\delta_\tau - r_1\delta_\theta = 0 \quad (1)$$

$$-\delta_v + u\delta_\tau - r_2\delta_\theta - \delta_s = 0 \quad (2)$$

$$-A_1^T\delta_y + \delta_w + c_1\delta_\tau - r_3\delta_\theta - \delta_h = 0 \quad (3)$$

$$-A_2^T\delta_y + c_2\delta_\tau - r_4\delta_\theta - \delta_z = 0 \quad (4)$$

$$b^T\delta_y - u^T\delta_w - c_1^T\delta_v - c_2^T\delta_x - r_5\delta_\theta - \delta_\kappa = 0 \quad (5)$$

$$r_1^T\delta_y + r_2^T\delta_w + r_3^T\delta_v + r_4^T\delta_x + r_5\delta_\tau = 0 \quad (6)$$

$$S\delta_w + W\delta_s = \mu e_1 - Ws \quad (7)$$

$$H\delta_v + V\delta_h = \mu e_2 - Vh \quad (8)$$

$$Z\delta_x + X\delta_z = \mu e_3 - Xz \quad (9)$$

$$\kappa\delta_\tau + \tau\delta_\kappa = \mu - \tau\kappa \quad (10)$$

onde e_1, e_2 e e_3 são vetores unitários de dimensão adequada.

A fim de resolvermos essa equação adotaremos a estratégia de isolar $\delta_y, \delta_w, \delta_v, \delta_x, \delta_s, \delta_h$ e δ_z em termos de $\delta_\tau, \delta_\theta$ e δ_κ , que são escalares, e substituir nas equações (5), (6) e (10), obtendo um sistema de três equações nas variáveis $\delta_\tau, \delta_\theta$ e δ_κ , o que permite que elas sejam escritas apenas em termos de μ ; em seguida substituímos as expressões de $\delta_\tau, \delta_\theta$ e δ_κ nas expressões de $\delta_y, \delta_w, \delta_v, \delta_x, \delta_s, \delta_h$ e δ_z , obtendo assim todos os passos exclusivamente em termos de μ .

Isolando δ_y

$$A_1\delta_v + A_2\delta_x - b\delta_\tau - r_1\delta_\theta = 0 \quad (1)$$

$$-\delta_v + u\delta_\tau - r_2\delta_\theta - \delta_s = 0 \quad (2)$$

$$-A_1^T\delta_y + \delta_w + c_1\delta_\tau - r_3\delta_\theta - \delta_h = 0 \quad (3)$$

$$-A_2^T\delta_y + c_2\delta_\tau - r_4\delta_\theta - \delta_z = 0 \quad (4)$$

$$b^T\delta_y - u^T\delta_w - c_1^T\delta_v - c_2^T\delta_x - r_5\delta_\theta - \delta_\kappa = 0 \quad (5)$$

$$r_1^T\delta_y + r_2^T\delta_w + r_3^T\delta_v + r_4^T\delta_x + r_5\delta_\tau = 0 \quad (6)$$

$$S\delta_w + W\delta_s = \mu e_1 - Ws \quad (7)$$

$$H\delta_v + V\delta_h = \mu e_2 - Vh \quad (8)$$

$$Z\delta_x + X\delta_z = \mu e_3 - Xz \quad (9)$$

$$\kappa\delta_\tau + \tau\delta_\kappa = \mu - \tau\kappa \quad (10)$$

Multiplicando-se a equação (7) por $S^{-1} = Q_1$, a equação (8) por $V^{-1} = Q_2$, a equação (9) por $X^{-1} = Q_3$, a equação (2) por $-S^{-1}W$ e substituindo

$$d_1 = S^{-1}W$$

$$d_2 = V^{-1}H$$

$$d_3 = X^{-1}Z$$

$$d_4 = (d_1 + d_2)^{-1}$$

$$d_5 = d_3^{-1}$$

e convencionando que

$$q_1 = S^{-1}e_1$$

$$q_2 = V^{-1}e_2$$

$$q_3 = X^{-1}e_3$$

obtemos

$$A_1\delta_v + A_2\delta_x - b\delta_\tau - r_1\delta_\theta = 0 \quad (1)$$

$$d_1\delta_v - d_1u\delta_\tau + d_1r_2\delta_\theta + d_1\delta_s = 0 \quad (2a)$$

$$-A_1^T\delta_y + \delta_w + c_1\delta_\tau - r_3\delta_\theta - \delta_h = 0 \quad (3)$$

$$-A_2^T\delta_y + c_2\delta_\tau - r_4\delta_\theta - \delta_z = 0 \quad (4)$$

$$b^T\delta_y - u^T\delta_w - c_1^T\delta_v - c_2^T\delta_x - r_5\delta_\theta - \delta_\kappa = 0 \quad (5)$$

$$r_1^T\delta_y + r_2^T\delta_w + r_3^T\delta_v + r_4^T\delta_x + r_5\delta_\tau = 0 \quad (6)$$

$$-\delta_w - d_1\delta_s + \mu q_1 - w = 0 \quad (7a)$$

$$d_2\delta_v + \delta_h - \mu q_2 + h = 0 \quad (8a)$$

$$d_3\delta_x + \delta_z - \mu q_3 + z = 0 \quad (9a)$$

$$\kappa\delta_\tau + \tau\delta_\kappa - \mu + \tau\kappa = 0 \quad (10)$$

Somando-se agora as equações (2a), (3), (7a), (8a) obtemos

$$\begin{aligned} & (d_1\delta_v - d_1u\delta_\tau + d_1r_2\delta_\theta + d_1\delta_s) + \\ & (-A_1^T\delta_y + \delta_w + c_1\delta_\tau - r_3\delta_\theta - \delta_h) + \\ & (-\delta_w - d_1\delta_s + \mu q_1 - w) + \\ & d_2\delta_v + \delta_h - \mu q_2 + h \\ & = 0 \end{aligned}$$

ou

$$(q_1 - q_2)\mu + (c_1 - d_1u)\delta_\tau + (d_1r_2 - r_3)\delta_\theta + (d_1 + d_2)\delta_v + h - w - A_1^T\delta_y = 0 \quad (3a)$$

Multiplicando-se agora a equação (3a) por $-A_1(d_1 + d_2)^{-1}$, somando-se à equação

(1) obtemos

$$\begin{aligned}
& (-A_1 d_4 (c_1 - d_1 u) - b) \delta_\tau + (-A_1 d_4 (d_1 r_2 - r_3) - r_1) \delta_\theta - \\
& A_1 d_4 (q_1 - q_2) \mu + A_1 d_4 A_1^T \delta_y + A_2 \delta_x - A_1 d_4 h + A_1 d_4 w \\
& = 0
\end{aligned}$$

Somando-se agora a equação (9a) à equação (4) obtemos

$$-A_2^T \delta_y + c_2 \delta_\tau - r_4 \delta_\theta + d_3 \delta_x - \mu q_3 + z = 0$$

Multiplicando-se essa equação por $-A_2 d_3^{-1}$, somando-se à equação anterior obtemos

$$\begin{aligned}
(A_1 d_4 A_1^T + A_2 d_5 A_2^T) \delta_y &= (A_1 d_4 (c_1 - d_1 u) + A_2 d_5 c_2 + b) \delta_\tau \\
&+ (A_1 d_4 (d_1 r_2 - r_3) - A_2 d_5 r_4 + r_1) \delta_\theta \\
&+ (A_2 d_5 q_3 - A_1 d_4 (q_1 - q_2)) \mu \\
&+ (A_1 d_4 (q_1 - q_2) - A_2 d_5 q_3)
\end{aligned}$$

Definindo agora

$$\begin{aligned}
B &= (A_1 d_4 A_1^T + A_2 d_5 A_2^T) \\
t_1 &= B^{-1} (A_1 d_4 (c_1 - d_1 u) + A_2 d_5 c_2 + b) \\
t_2 &= B^{-1} (A_1 d_4 (d_1 r_2 - r_3) - A_2 d_5 r_4 + r_1) \\
t_3 &= B^{-1} (A_1 d_4 (q_1 - q_2) - A_2 d_5 q_3) \\
t_4 &= B^{-1} (A_1 d_4 (h - w) + A_2 d_5 z)
\end{aligned}$$

obtemos finalmente

$$\delta_y = t_1 \delta_\tau + t_2 \delta_\theta + t_3 \mu + t_4$$

Observe que os vetores t_1, t_2, t_3 e t_4 são obtidos pela resolução de quatro sistemas lineares que utilizam a mesma fatoração de Cholesky de B . Esse sistema, como tínhamos afirmado, é do mesmo tamanho dos sistemas lineares usuais dos algoritmos primal-dual.

Isolando δ_v

Multiplicando-se a equação (3a) por d_4 obtemos

$$d_4 ((q_1 - q_2) \mu + (c_1 - d_1 u) \delta_\tau + (d_1 r_2 - r_3) \delta_\theta + (d_1 + d_2) \delta_v + h - w - A_1^T \delta_y) = 0$$

substituindo δ_y obtemos

$$\begin{aligned} \delta_v = & d_4 (A_1^T t_1 + d_1 u - c_1) \delta_\tau + d_4 (A_1^T t_2 - d_1 r_2 + r_3) \delta_\theta + \\ & d_4 (A_1^T t_3 + q_2 - q_1) \mu + d_4 (A_1^T t_4 + w - h) \end{aligned}$$

e definindo

$$f_1 = A_1^T t_1$$

$$f_2 = A_1^T t_2$$

$$f_3 = A_1^T t_3$$

$$f_4 = A_1^T t_4$$

$$f_5 = A_2^T t_1$$

$$f_6 = A_2^T t_2$$

$$f_7 = A_2^T t_3$$

$$f_8 = A_2^T t_4$$

e

$$t_5 = d_4 (f_1 + d_1 u - c_1)$$

$$t_6 = d_4 (f_2 - d_1 r_2 + r_3)$$

$$t_7 = d_4 (f_3 + q_2 - q_1)$$

$$t_8 = d_4 (f_4 + w - h)$$

obtemos

$$\delta_v = t_5 \delta_\tau + t_6 \delta_\theta + t_7 \mu + t_8$$

Isolando δ_s

A partir da equação (2) :

$$-\delta_v + u \delta_\tau - r_2 \delta_\theta - \delta_s = 0$$

e substituindo-se δ_v obtemos

$$\delta_s = (-t_5 + u) \delta_\tau + (-t_6 - r_2) \delta_\theta - t_7 \mu - t_8$$

Isolando δ_w

A partir da equação (7a) :

$$-\delta_w - d_1 \delta_s + \mu q_1 - w = 0$$

e substituindo δ_s , obtemos

$$\delta_w = d_1 (t_5 - u) \delta_\tau + d_1 (r_2 + t_6) \delta_\theta + (d_1 t_7 + q_1) \mu + d_1 t_8 - w$$

Isolando δ_z

A partir da equação (4) e substituindo δ_y obtemos

$$\delta_z = (c_2 - f_5) \delta_\tau + (-f_6 - r_4) \delta_\theta - f_7 \mu - f_8$$

Isolando δ_x

A partir da equação (9a):

$$d_3 \delta_x + \delta_z - \mu q_3 + z = 0$$

e multiplicando-se por $d_5 = d_3^{-1}$ obtemos

$$d_5 (d_3^{-1} \delta_x + \delta_z - \mu q_3 + z) = 0$$

e substituindo δ_z obtemos

$$\delta_x = d_5 (f_5 - c_2) \delta_\tau + d_5 (f_6 + r_4) \delta_\theta + d_5 (f_7 + q_3) \mu + d_5 f_8 - x$$

Isolando δ_h

A partir da equação (8a):

$$d_2 \delta_v + \delta_h - \mu q_2 + h = 0$$

e substituindo δ_v obtemos

$$\delta_h = -d_2 t_5 \delta_\tau - d_2 t_6 \delta_\theta + (q_2 - d_2 t_7) \mu - d_2 t_8 - h$$

Cálculo de δ_τ , δ_θ , e δ_κ em função de μ

A partir das equações (5), (6) e (10) substituindo $\delta_y, \delta_w, \delta_v$ e δ_x obtemos um sistema de três equações nas variáveis reais δ_τ , δ_θ , e δ_κ , o que nos permite expressá-las em termos de μ e, conseqüentemente, expressar os demais passos em termos de μ .

$$b^T \delta_y - u^T \delta_w - c_1^T \delta_v - c_2^T \delta_x - r_5 \delta_\theta - \delta_\kappa = 0 \quad (5)$$

$$r_1^T \delta_y + r_2^T \delta_w + r_3^T \delta_v + r_4^T \delta_x + r_5 \delta_\tau = 0 \quad (6)$$

$$\kappa \delta_\tau + \tau \delta_\kappa = \mu - \tau \kappa \quad (10)$$

Da equação 5 temos

$$b^T \delta_y - u^T \delta_w - c_1^T \delta_v - c_2^T \delta_x - r_5 \delta_\theta - \delta_\kappa = 0$$

ou

$$\begin{aligned} & (b^T t_1 - u^T d_1 (t_5 - u) - c_1^T t_5 - c_2^T d_5 (f_5 - c_2)) \delta_\tau \\ & + (b^T t_2 - u^T d_1 (r_2 + t_6) - c_1^T t_6 - c_2^T d_5 (f_6 + r_4) - r_5) \delta_\theta \\ & - \delta_\kappa \\ & + (b^T t_3 - u^T (d_1 t_7 + q_1) - c_1^T t_7 - c_2^T d_5 (f_7 + q_3)) \mu \\ & + b^T t_4 - u^T (d_1 t_8 - w) - c_1^T t_8 - c_2^T d_5 (f_8 - z) \\ & = 0 \end{aligned}$$

Da equação 6 temos

$$r_1^T \delta_y + r_2^T \delta_w + r_3^T \delta_v + r_4^T \delta_x + r_5 \delta_\tau = 0$$

ou

$$\begin{aligned}
& (r_1^T t_1 + r_2^T d_1 (t_5 - u) + r_3^T t_5 + r_4^T d_5 (f_5 - c_2) + r_5) \delta_\tau + \\
& (r_1^T t_2 + r_2^T d_1 (t_6 + r_2) + r_3^T t_6 + r_4^T d_5 (f_6 + r_4)) \delta_\theta + \\
& (r_1^T t_3 + r_2^T (d_1 t_7 + q_1) + r_3^T t_7 + r_4^T d_5 (f_7 + q_3)) \mu + \\
& r_1^T t_4 + r_2^T (d_1 t_8 - w) + r_3^T t_8 + r_4^T d_5 (f_8 - z) \\
& = 0
\end{aligned}$$

Da equação 10 temos

$$\kappa \delta_\tau + \tau \delta_\kappa - \mu + \tau \kappa = 0$$

Substituindo agora

$$\begin{aligned}
\alpha_1 &= b^T t_1 - u^T d_1 (t_5 - u) - c_1^T t_5 - c_2^T d_5 (f_5 - c_2) \\
\alpha_2 &= b^T t_2 - u^T d_1 (r_2 + t_6) - c_1^T t_6 - c_2^T d_5 (f_6 + r_4) - r_5 \\
\alpha_3 &= b^T t_3 - u^T (d_1 t_7 + q_1) - c_1^T t_7 - c_2^T d_5 (f_7 + q_3) \\
\alpha_4 &= b^T t_4 - u^T (d_1 t_8 - w) - c_1^T t_8 - c_2^T d_5 (f_8 - z) \\
\alpha_5 &= r_1^T t_1 + r_2^T d_1 (t_5 - u) + r_3^T t_5 + r_4^T d_5 (f_5 - c_2) + r_5 \\
\alpha_6 &= r_1^T t_2 + r_2^T d_1 (t_6 + r_2) + r_3^T t_6 + r_4^T d_5 (f_6 + r_4) \\
\alpha_7 &= r_1^T t_3 + r_2^T (d_1 t_7 + q_1) + r_3^T t_7 + r_4^T d_5 (f_7 + q_3) \\
\alpha_8 &= r_1^T t_4 + r_2^T (d_1 t_8 - w) + r_3^T t_8 + r_4^T d_5 (f_8 - z)
\end{aligned}$$

e junto com a equação (10) : $\kappa \delta_\tau + \tau \delta_\kappa - \mu + \tau \kappa = 0$ obtemos o sistema

$$\begin{aligned}
\alpha_1 \delta_\tau + \alpha_2 \delta_\theta - \delta_\kappa &= -\alpha_3 \mu - \alpha_4 \\
\alpha_5 \delta_\tau + \alpha_6 \delta_\theta &= -\alpha_7 \mu - \alpha_8 \\
\kappa \delta_\tau + \tau \delta_\kappa &= \mu - \tau \kappa
\end{aligned}$$

cuja solução é

$$\delta_\kappa = \frac{-\kappa\alpha_6\alpha_3 - \alpha_6\alpha_1 + \alpha_2\alpha_5 + \kappa\alpha_2\alpha_7}{-\alpha_6\tau\alpha_1 - \alpha_6\rho\kappa + \tau\alpha_2\alpha_5}\mu + \frac{-\kappa\alpha_6\alpha_4 + \tau\kappa\alpha_6\alpha_1 + \kappa\alpha_2\alpha_8 - \tau\kappa\alpha_2\alpha_5}{-\alpha_6\tau\alpha_1 - \alpha_6\rho\kappa + \tau\alpha_2\alpha_5}$$

$$\delta_\kappa = \frac{\alpha_6\alpha_1 + \alpha_6\kappa\alpha_3 - \alpha_2\alpha_5 - \alpha_2\kappa\alpha_7}{\kappa\alpha_6 + \alpha_6\alpha_1\tau - \alpha_2\alpha_5\tau}\mu + \frac{\alpha_6\kappa\alpha_4 - \alpha_2\kappa\alpha_8 - \alpha_6\alpha_1\tau\kappa + \alpha_2\alpha_5\tau\kappa}{\kappa\alpha_6 + \alpha_6\alpha_1\tau - \alpha_2\alpha_5\tau}$$

$$\delta_\theta = -\frac{-\alpha_5\rho + \alpha_5\tau\alpha_3 - \alpha_7\tau\alpha_1 - \alpha_7\rho\kappa}{-\alpha_6\tau\alpha_1 - \alpha_6\rho\kappa + \tau\alpha_2\alpha_5}\mu - \frac{\alpha_5\tau\alpha_4 + \alpha_5\rho\tau\kappa - \alpha_8\tau\alpha_1 - \alpha_8\rho\kappa}{-\alpha_6\tau\alpha_1 - \alpha_6\rho\kappa + \tau\alpha_2\alpha_5}$$

$$\delta_\theta = -\frac{\alpha_5 + \kappa\alpha_7 + \alpha_7\alpha_1\tau - \tau\alpha_5\alpha_3}{\kappa\alpha_6 + \alpha_6\alpha_1\tau - \alpha_2\alpha_5\tau}\mu - \frac{\kappa\alpha_8 + \alpha_8\alpha_1\tau - \tau\alpha_5\alpha_4 - \alpha_5\tau\kappa}{\kappa\alpha_6 + \alpha_6\alpha_1\tau - \alpha_2\alpha_5\tau}$$

$$\delta_\tau = -\frac{\tau\alpha_2\alpha_7 + \alpha_6\rho - \alpha_6\tau\alpha_3}{-\alpha_6\tau\alpha_1 - \alpha_6\rho\kappa + \tau\alpha_2\alpha_5}\mu - \frac{\tau\alpha_2\alpha_8 - \alpha_6\rho\tau\kappa - \alpha_6\tau\alpha_4}{-\alpha_6\tau\alpha_1 - \alpha_6\rho\kappa + \tau\alpha_2\alpha_5}$$

$$\delta_\tau = \frac{\tau\alpha_2\alpha_7 + \alpha_6 - \tau\alpha_6\alpha_3}{\kappa\alpha_6 + \alpha_6\alpha_1\tau - \alpha_2\alpha_5\tau}\mu + \frac{\tau\alpha_2\alpha_8 - \tau\alpha_6\alpha_4 - \tau\kappa\alpha_6}{\kappa\alpha_6 + \alpha_6\alpha_1\tau - \alpha_2\alpha_5\tau}$$

Se definirmos

$$\begin{aligned} p_1 &= \frac{\alpha_6\alpha_1 + \alpha_6\kappa\alpha_3 - \alpha_2\alpha_5 - \alpha_2\kappa\alpha_7}{\kappa\alpha_6 + \alpha_6\alpha_1\tau - \alpha_2\alpha_5\tau} \\ p_2 &= \frac{\alpha_6\kappa\alpha_4 - \alpha_2\kappa\alpha_8 - \alpha_6\alpha_1\tau\kappa + \alpha_2\alpha_5\tau\kappa}{\kappa\alpha_6 + \alpha_6\alpha_1\tau - \alpha_2\alpha_5\tau} \\ p_3 &= -\frac{\alpha_5 + \kappa\alpha_7 + \alpha_7\alpha_1\tau - \tau\alpha_5\alpha_3}{\kappa\alpha_6 + \alpha_6\alpha_1\tau - \alpha_2\alpha_5\tau} \\ p_4 &= -\frac{\kappa\alpha_8 + \alpha_8\alpha_1\tau - \tau\alpha_5\alpha_4 - \alpha_5\tau\kappa}{\kappa\alpha_6 + \alpha_6\alpha_1\tau - \alpha_2\alpha_5\tau} \\ p_5 &= \frac{\tau\alpha_2\alpha_7 + \alpha_6 - \tau\alpha_6\alpha_3}{\kappa\alpha_6 + \alpha_6\alpha_1\tau - \alpha_2\alpha_5\tau} \\ p_6 &= \frac{\tau\alpha_2\alpha_8 - \tau\alpha_6\alpha_4 - \tau\kappa\alpha_6}{\kappa\alpha_6 + \alpha_6\alpha_1\tau - \alpha_2\alpha_5\tau} \end{aligned}$$

então

$$\delta_\kappa = p_1\mu + p_2$$

$$\delta_\theta = p_3\mu + p_4$$

$$\delta_\tau = p_5\mu + p_6$$

Cálculo de $\delta_v, \delta_h, \delta_x, \delta_z, \delta_w$ e δ_s em função de μ

Devemos substituir os valores que acabamos de obter de $\delta_\tau, \delta_\theta$, e δ_κ nas equações

$$\delta_y = t_1\delta_\tau + t_2\delta_\theta + t_3\mu + t_4$$

$$\delta_v = t_5\delta_\tau + t_6\delta_\theta + t_7\mu + t_8$$

$$\delta_h = -d_2t_5\delta_\tau - d_2t_6\delta_\theta + (-d_2t_7 + q_2)\mu - d_2t_8 - h$$

$$\delta_x = d_5(f_5 - c_2)\delta_\tau + d_5(f_6 + r_4)\delta_\theta + d_5(f_7 + q_3)\mu + d_5(f_8 - z)$$

$$\delta_z = (c_2 - f_5)\delta_\tau + (-f_6 - r_4)\delta_\theta - f_7\mu - f_8$$

$$\delta_w = d_1(t_5 - u)\delta_\tau + d_1(r_2 + t_6)\delta_\theta + (d_1t_7 + q_1)\mu + d_1t_8 - w$$

$$\delta_s = (-t_5 + u)\delta_\tau + (-t_6 - r_2)\delta_\theta - t_7\mu - t_8$$

isto é,

$$\delta_y = (t_2p_3 + t_3 + t_1p_5)\mu + t_1p_6 + t_4 + t_2p_4$$

$$\delta_v = (t_6p_3 + t_7 + t_5p_5)\mu + t_5p_6 + t_8 + t_6p_4$$

$$\delta_h = (q_2 - d_2(t_6p_3 + t_7 + t_5p_5))\mu - d_2(t_5p_6 + t_8 + t_6p_4) - h$$

$$\delta_x = d_5((f_6 + r_4)p_3 + f_7 + q_3 + (f_5 - c_2)p_5)\mu + d_5((f_5 - c_2)p_6 + f_8 - z + (f_6 + r_4)p_4)$$

$$\delta_z = ((-f_6 - r_4)p_3 - f_7 + (c_2 - f_5)p_5)\mu + (c_2 - f_5)p_6 - f_8 + (-f_6 - r_4)p_4$$

$$\delta_w = (q_1 + d_1((r_2 + t_6)p_3 + t_7 + (t_5 - u)p_5))\mu + d_1((t_5 - u)p_6 + t_8 + (r_2 + t_6)p_4) - w$$

$$\delta_s = ((-t_6 - r_2)p_3 - t_7 + (-t_5 + u)p_5)\mu + (-t_5 + u)p_6 - t_8 + (-t_6 - r_2)p_4$$

Para simplificar as expressões que se seguem definimos

$$\begin{aligned}
\gamma_1 &= t_2 p_3 + t_3 + t_1 p_5 \\
\gamma_2 &= t_1 p_6 + t_4 + t_2 p_4 \\
\gamma_3 &= t_6 p_3 + t_7 + t_5 p_5 \\
\gamma_4 &= t_5 p_6 + t_8 + t_6 p_4 \\
\gamma_5 &= q_2 - d_2 (t_6 p_3 + t_7 + t_5 p_5) \\
\gamma_6 &= -d_2 (t_5 p_6 + t_8 + t_6 p_4) - h \\
\gamma_7 &= d_5 ((f_6 + r_4) p_3 + f_7 + q_3 + (f_5 - c_2) p_5) \\
\gamma_8 &= d_5 ((f_5 - c_2) p_6 + f_8 - z + (f_6 + r_4) p_4) \\
\gamma_9 &= (-f_6 - r_4) p_3 - f_7 + (c_2 - f_5) p_5 \\
\gamma_{10} &= (c_2 - f_5) p_6 - f_8 + (-f_6 - r_4) p_4 \\
\gamma_{11} &= q_1 + d_1 ((r_2 + t_6) p_3 + t_7 + (t_5 - u) p_5) \\
\gamma_{12} &= d_1 ((t_5 - u) p_6 + t_8 + (r_2 + t_6) p_4) - w \\
\gamma_{13} &= (-t_6 - r_2) p_3 - t_7 + (-t_5 + u) p_5 \\
\gamma_{14} &= (-t_5 + u) p_6 - t_8 + (-t_6 - r_2) p_4
\end{aligned}$$

de modo que

$$\delta_y = \gamma_1\mu + \gamma_2$$

$$\delta_v = \gamma_3\mu + \gamma_4$$

$$\delta_h = \gamma_5\mu + \gamma_6$$

$$\delta_x = \gamma_7\mu + \gamma_8$$

$$\delta_z = \gamma_9\mu + \gamma_{10}$$

$$\delta_w = \gamma_{11}\mu + \gamma_{12}$$

$$\delta_s = \gamma_{13}\mu + \gamma_{14}$$

$$\delta_\kappa = p_1\mu + p_2$$

$$\delta_\theta = p_3\mu + p_4$$

$$\delta_\tau = p_5\mu + p_6$$

A.2 Parametrização de segunda ordem

No caso do contexto auto-dual com variáveis canalizadas, o sistema linear que precisamos resolver, equivalente ao sistema 3.1 no contexto primal-dual, é dado por

$$A_1\Delta_v + A_2\Delta_x - b\Delta_\tau - r_1\Delta_\theta = 0 \quad (1)$$

$$-\Delta_v + u\Delta_\tau - r_2\Delta_\theta - \Delta_s = 0 \quad (2)$$

$$-A_1^T\Delta_y + \Delta_w + c_1\Delta_\tau - r_3\Delta_\theta - \Delta_h = 0 \quad (3)$$

$$-A_2^T\Delta_y + c_2\Delta_\tau - r_4\Delta_\theta - \Delta_z = 0 \quad (4)$$

$$b^T\Delta_y - u^T\Delta_w - c_1^T\Delta_v - c_2^T\Delta_x - r_5\Delta_\theta - \Delta_\kappa = 0 \quad (5)$$

$$r_1^T\Delta_y + r_2^T\Delta_w + r_3^T\Delta_v + r_4^T\Delta_x + r_5\Delta_\tau = 0 \quad (6)$$

$$S\Delta_w + W\Delta_s = \mu e_1 - Ws - \delta_s\delta_w \quad (7)$$

$$H\Delta_v + V\Delta_h = \mu e_2 - Vh - \delta_h\delta_v \quad (8)$$

$$Z\Delta_x + X\Delta_z = \mu e_3 - Xz - \delta_x\delta_z \quad (9)$$

$$\kappa\Delta_\tau + \tau\Delta_\kappa = \mu - \tau\kappa - \delta_\kappa\delta_\tau \quad (10)$$

onde e_1, e_2 e e_3 são vetores unitários de dimensão adequada, e portanto de acordo com A.1 temos

$$\delta_s\delta_w = (\gamma_{13}\mu + \gamma_{14})(\gamma_{11}\mu + \gamma_{12})$$

$$\delta_h\delta_v = (\gamma_5\mu + \gamma_6)(\gamma_3\mu + \gamma_4)$$

$$\delta_x\delta_z = (\gamma_9\mu + \gamma_{10})(\gamma_7\mu + \gamma_8)$$

$$\delta_\kappa\delta_\tau = (p_1\mu + p_2)(p_5\mu + p_6)$$

Afim de resolvermos essa equação adotaremos a estratégia de isolar $\Delta_y, \Delta_w, \Delta_v, \Delta_x, \Delta_s, \Delta_h$ e Δ_z em termos de $\Delta_\tau, \Delta_\theta$ e Δ_κ , que são escalares, e substituir nas equações (5), (6) e (10), obtendo um sistema de três equações nas variáveis $\Delta_\tau, \Delta_\theta$ e Δ_κ , o que permite que elas sejam escritas apenas em termos de μ ; em seguida substituímos as expressões de $\Delta_\tau, \Delta_\theta$ e Δ_κ nas expressões de $\Delta_y, \Delta_w, \Delta_v, \Delta_x, \Delta_s, \Delta_h$ e Δ_z , obtendo assim todos os

passos exclusivamente em termos de μ .

Isolando Δ_y

Multiplicando-se a equação (7) por $-S^{-1} = -Q_1$, a equação (8) por $V^{-1} = Q_2$, a equação (9) por $X^{-1} = Q_3$, a equação (2) por $-S^{-1}W$ e substituindo

$$D_1 = S^{-1}W$$

$$D_2 = V^{-1}H$$

$$D_3 = X^{-1}Z$$

$$D_4 = (D_1 + D_2)^{-1}$$

$$D_5 = D_3^{-1}$$

e convencionando que

$$q_1 = S^{-1}e_1$$

$$q_2 = V^{-1}e_2$$

$$q_3 = X^{-1}e_3$$

obtemos

$$A_1 \Delta_v + A_2 \Delta_x - b \Delta_\tau - r_1 \Delta_\theta = 0 \quad (1)$$

$$D_1 \Delta_v - D_1 u \Delta_\tau + D_1 r_2 \Delta_\theta + D_1 \Delta_s = 0 \quad (2a)$$

$$-A_1^T \Delta_y + \Delta_w + c_1 \Delta_\tau - r_3 \Delta_\theta - \Delta_h = 0 \quad (3)$$

$$-A_2^T \Delta_y + c_2 \Delta_\tau - r_4 \Delta_\theta - \Delta_z = 0 \quad (4)$$

$$b^T \Delta_y - u^T \Delta_w - c_1^T \Delta_v - c_2^T \Delta_x - r_5 \Delta_\theta - \Delta_\kappa = 0 \quad (5)$$

$$r_1^T \Delta_y + r_2^T \Delta_w + r_3^T \Delta_v + r_4^T \Delta_x + r_5 \Delta_\tau = 0 \quad (6)$$

$$-\Delta_w - D_1 \Delta_s + \mu q_1 - w - Q_1 \delta_s \delta_w = 0 \quad (7a)$$

$$D_2 \Delta_v + \Delta_h - \mu q_2 + h + Q_2 \delta_h \delta_v = 0 \quad (8a)$$

$$D_3 \Delta_x + \Delta_z - \mu q_3 + z + Q_3 \delta_x \delta_z = 0 \quad (9a)$$

$$\kappa \Delta_\tau + \tau \Delta_\kappa - \mu + \tau \kappa + \delta_\kappa \delta_\tau = 0 \quad (10)$$

Somando-se agora as equações (2a), (3), (7a), (8a)

$$D_1 \Delta_v - D_1 u \Delta_\tau + D_1 r_2 \Delta_\theta + D_1 \Delta_s = 0 \quad (2a)$$

$$-A_1^T \Delta_y + \Delta_w + c_1 \Delta_\tau - r_3 \Delta_\theta - \Delta_h = 0 \quad (3)$$

$$-\Delta_w - D_1 \Delta_s + \mu q_1 - w - Q_1 \delta_s \delta_w = 0 \quad (7a)$$

$$D_2 \Delta_v + \Delta_h - \mu q_2 + h + Q_2 \delta_h \delta_v = 0 \quad (8a)$$

obtemos

$$\begin{aligned} & (q_1 - q_2) \mu + (c_1 - D_1 u) \Delta_\tau + (D_1 r_2 - r_3) \Delta_\theta + (D_1 + D_2) \Delta_v \quad (3a) \\ & + h - w - A_1^T \Delta_y - Q_1 \delta_s \delta_w + Q_2 \delta_h \delta_v \\ & = 0 \end{aligned}$$

Multiplicando-se agora a equação (3a) por $-A_1 (D_1 + D_2)^{-1}$, somando-se à equação

(1) obtemos

$$\begin{aligned}
& (-A_1 D_4 (c_1 - D_1 u) - b) \Delta_\tau + (-A_1 D_4 (D_1 r_2 - r_3) - r_1) \Delta_\theta - \\
& A_1 D_4 (q_1 - q_2) \mu + A_1 D_4 A_1^T \Delta_y + A_2 \Delta_x - A_1 D_4 h + A_1 D_4 w + \\
& A_1 D_4 (Q_1 \delta_s \delta_w - Q_2 \delta_h \delta_v) \\
& = 0
\end{aligned}$$

Somando-se agora a equação (9a) à equação (4) obtemos

$$-A_2^T \Delta_y + c_2 \Delta_\tau - r_4 \Delta_\theta + D_3 \Delta_x - \mu q_3 + z + Q_3 \delta_x \delta_z = 0$$

Multiplicando-se essa equação por $-A_2 D_3^{-1}$, somando-se à equação anterior obtemos

$$\begin{aligned}
(A_1 D_4 A_1^T + A_2 D_5 A_2^T) \Delta_y &= (A_1 D_4 (c_1 - D_1 u) + A_2 D_5 c_2 + b) \Delta_\tau \\
&+ (A_1 D_4 (D_1 r_2 - r_3) - A_2 D_5 r_4 + r_1) \Delta_\theta \\
&+ (A_1 D_4 (q_1 - q_2) - A_2 D_5 q_3) \mu \\
&+ A_1 D_4 (h - w) + A_2 D_5 z \\
&- A_1 D_4 (Q_1 \delta_s \delta_w - Q_2 \delta_h \delta_v) + A_2 D_5 Q_3 \delta_x \delta_z
\end{aligned}$$

$$B = (A_1 d_4 A_1^T + A_2 d_5 A_2^T)$$

$$t_1 = B^{-1} (A_1 d_4 (c_1 - d_1 u) + A_2 d_5 c_2 + b)$$

$$t_2 = B^{-1} (A_1 d_4 (d_1 r_2 - r_3) - A_2 d_5 r_4 + r_1)$$

$$t_3 = B^{-1} (A_1 d_4 (q_1 - q_2) - A_2 d_5 q_3)$$

$$t_4 = B^{-1} (A_1 d_4 (h - w) + A_2 d_5 z)$$

Definindo agora

$$\begin{aligned}
B &= (A_1 D_4 A_1^T + A_2 D_5 A_2^T) \\
t_1 &= B^{-1} (A_1 D_4 (c_1 - D_1 u) + A_2 D_5 c_2 + b) \\
t_2 &= B^{-1} (A_1 D_4 (D_1 r_2 - r_3) - A_2 D_5 r_4 + r_1) \\
t_3 &= B^{-1} (A_1 D_4 (q_1 - q_2) - A_2 D_5 q_3) \\
t_4 &= B^{-1} (A_1 D_4 (h - w) + A_2 D_5 z) \\
t_{ry} &= B^{-1} (A_1 D_4 (-Q_1 \delta_s \delta_w + Q_2 \delta_h \delta_v) + A_2 D_5 Q_3 \delta_x \delta_z) \\
&= \eta_2 \mu^2 + \eta_1 \mu + \eta_0
\end{aligned}$$

obtemos finalmente

$$\Delta_y = t_1 \Delta_\tau + t_2 \Delta_\theta + t_3 \mu + t_4 + t_{ry}.$$

Observe que a matriz B e sua fatoração de Cholesky, bem como os vetores t_1, t_2, t_3 e t_4 , já foram todos calculados anteriormente. Mostraremos a seguir que o cálculo de t_{ry} envolve a resolução de mais três sistemas lineares, e com isso a cada passo no contexto auto-dual temos que resolver 7 sistemas lineares.

Cálculo de η

Como

$$\begin{aligned}
\delta_s \delta_w &= \gamma_{11} \gamma_{13} \mu^2 + (\gamma_{12} \gamma_{13} + \gamma_{11} \gamma_{14}) \mu + \gamma_{12} \gamma_{14} \\
\delta_h \delta_v &= \gamma_3 \gamma_5 \mu^2 + (\gamma_4 \gamma_5 + \gamma_3 \gamma_6) \mu + \gamma_4 \gamma_6 \\
\delta_x \delta_z &= \gamma_7 \gamma_9 \mu^2 + (\gamma_8 \gamma_9 + \gamma_7 \gamma_{10}) \mu + \gamma_8 \gamma_{10} \\
\delta_\kappa \delta_\tau &= p_5 p_1 \mu^2 + (p_6 p_1 + p_5 p_2) \mu + p_6 p_2
\end{aligned}$$

podemos calcular

$$\begin{aligned}
t_{ry} &= B^{-1} (A_1 D_4 (-Q_1 \delta_s \delta_w + Q_2 \delta_h \delta_v) + A_2 D_5 Q_3 \delta_x \delta_z) \\
&= B^{-1} (A_1 D_4 (-Q_1 \gamma_{11} \gamma_{13} + Q_2 \gamma_3 \gamma_5) + A_2 D_5 Q_3 \gamma_7 \gamma_9) \mu^2 \\
&\quad + B^{-1} (A_1 D_4 (-Q_1 (\gamma_{12} \gamma_{13} + \gamma_{11} \gamma_{14}) + Q_2 (\gamma_4 \gamma_5 + \gamma_3 \gamma_6)) + A_2 D_5 Q_3 (\gamma_8 \gamma_9 + \gamma_7 \gamma_{10})) \mu \\
&\quad + B^{-1} (A_1 D_4 (-Q_1 \gamma_{12} \gamma_{14} + Q_2 \gamma_4 \gamma_6) + A_2 D_5 Q_3 \gamma_8 \gamma_{10})
\end{aligned}$$

e assim temos

$$\begin{aligned}
\eta_2 &= B^{-1} (A_1 D_4 (-Q_1 \gamma_{11} \gamma_{13} + Q_2 \gamma_3 \gamma_5) + A_2 D_5 Q_3 \gamma_7 \gamma_9) \\
&= B^{-1} \begin{bmatrix} A_1 & A_2 \end{bmatrix} \begin{bmatrix} D_4 & 0 \\ 0 & D_5 \end{bmatrix} \begin{bmatrix} -Q_1 \gamma_{11} \gamma_{13} + Q_2 \gamma_3 \gamma_5 \\ Q_3 \gamma_7 \gamma_9 \end{bmatrix} \\
&= B^{-1} \begin{bmatrix} A_1 & A_2 \end{bmatrix} \begin{bmatrix} D_4 & 0 \\ 0 & D_5 \end{bmatrix} \begin{bmatrix} -S^{-1} \gamma_{11} \gamma_{13} + V^{-1} \gamma_3 \gamma_5 \\ X^{-1} \gamma_7 \gamma_9 \end{bmatrix} \\
\eta_1 &= B^{-1} (A_1 D_4 (-Q_1 (\gamma_{12} \gamma_{13} + \gamma_{11} \gamma_{14}) + Q_2 (\gamma_4 \gamma_5 + \gamma_3 \gamma_6)) + A_2 D_5 Q_3 (\gamma_8 \gamma_9 + \gamma_7 \gamma_{10})) \\
&= B^{-1} \begin{bmatrix} A_1 & A_2 \end{bmatrix} \begin{bmatrix} D_4 & 0 \\ 0 & D_5 \end{bmatrix} \begin{bmatrix} -Q_1 (\gamma_{12} \gamma_{13} + \gamma_{11} \gamma_{14}) + Q_2 (\gamma_4 \gamma_5 + \gamma_3 \gamma_6) \\ Q_3 (\gamma_8 \gamma_9 + \gamma_7 \gamma_{10}) \end{bmatrix} \\
\eta_0 &= B^{-1} (A_1 D_4 (-Q_1 \gamma_{12} \gamma_{14} + Q_2 \gamma_4 \gamma_6) + A_2 D_5 Q_3 \gamma_8 \gamma_{10}) \\
&= B^{-1} \begin{bmatrix} A_1 & A_2 \end{bmatrix} \begin{bmatrix} D_4 & 0 \\ 0 & D_5 \end{bmatrix} \begin{bmatrix} -Q_1 \gamma_{12} \gamma_{14} + Q_2 \gamma_4 \gamma_6 \\ Q_3 \gamma_8 \gamma_{10} \end{bmatrix}.
\end{aligned}$$

No cálculo acima tivemos que efetuar mais três resoluções de sistema, obtendo assim um total de 7 sistemas.

Isolando Δ_v

Multiplicando-se a equação (3a) por D_4 obtemos

$$\begin{aligned} & D_4 ((q_1 - q_2) \mu + (c_1 - D_1 u) \Delta_\tau + (D_1 r_2 - r_3) \Delta_\theta + (D_1 + D_2) \Delta_v + h - w - A_1^T \Delta_y) \\ & + D_4 (-Q_1 \delta_s \delta_w + Q_2 \delta_h \delta_v) \\ & = 0 \end{aligned}$$

ou substituindo Δ_y obtemos

$$\begin{aligned} \Delta_v &= D_4 (A_1^T t_1 + D_1 u - c_1) \Delta_\tau \\ &+ D_4 (A_1^T t_2 - D_1 r_2 + r_3) \Delta_\theta \\ &+ D_4 (Q_1 \gamma_{11} \gamma_{13} - Q_2 \gamma_3 \gamma_5 + A_1^T \eta_2) \mu^2 \\ &+ D_4 (-q_1 + Q_1 \gamma_{12} \gamma_{13} + A_1^T \eta_1 - Q_2 \gamma_4 \gamma_5 + Q_1 \gamma_{11} \gamma_{14} + A_1^T t_3 - Q_2 \gamma_3 \gamma_6 + q_2) \mu \\ &+ D_4 (Q_1 \gamma_{12} \gamma_{14} + A_2^T \eta_0 + w + A_1^T t_4 - Q_2 \gamma_4 \gamma_6 - h) \end{aligned}$$

e definindo

$$f_1 = A_1^T t_1$$

$$f_2 = A_1^T t_2$$

$$f_3 = A_1^T t_3$$

$$f_4 = A_1^T t_4$$

$$f_5 = A_2^T t_1$$

$$f_6 = A_2^T t_2$$

$$f_7 = A_2^T t_3$$

$$f_8 = A_2^T t_4$$

$$\begin{aligned}
\sigma_2 &= A_1^T \eta_2 \\
\sigma_1 &= A_1^T \eta_1 \\
\sigma_0 &= A_1^T \eta_0 \\
\sigma_3 &= A_2^T \eta_0 \\
\sigma_4 &= A_2^T \eta_1 \\
\sigma_5 &= A_2^T \eta_2
\end{aligned}$$

$$\begin{aligned}
t_5 &= D_4 (f_1 + D_1 u - c_1) \\
t_6 &= D_4 (f_2 - D_1 r_2 + r_3) \\
\nu_2 &= D_4 (Q_1 \gamma_{11} \gamma_{13} - Q_2 \gamma_3 \gamma_5 + \sigma_2) \\
\nu_1 &= D_4 (-q_1 + Q_1 \gamma_{12} \gamma_{13} + \sigma_1 - Q_2 \gamma_4 \gamma_5 + Q_1 \gamma_{11} \gamma_{14} + f_3 - Q_2 \gamma_3 \gamma_6 + q_2) \\
\nu_0 &= D_4 (Q_1 \gamma_{12} \gamma_{14} + \sigma_0 + w + f_4 - Q_2 \gamma_4 \gamma_6 - h)
\end{aligned}$$

obtemos

$$\Delta_v = t_5 \Delta_\tau + t_6 \Delta_\theta + \nu_2 \mu^2 + \nu_1 \mu + \nu_0$$

Isolando Δ_s

A partir da equação (2) :

$$-\Delta_v + u \Delta_\tau - r_2 \Delta_\theta - \Delta_s = 0$$

e substituindo-se Δ_v obtemos

$$\Delta_s = (u - t_5) \Delta_\tau + (-t_6 - r_2) \Delta_\theta - \nu_2 \mu^2 - \nu_1 \mu - \nu_0$$

Isolando Δ_w

A partir da equação (7a) :

$$-\Delta_w - D_1\Delta_s + \mu q_1 - w - Q_1\delta_s\delta_w = 0$$

e como

$$\delta_s\delta_w = \gamma_{11}\gamma_{13}\mu^2 + (\gamma_{12}\gamma_{13} + \gamma_{11}\gamma_{14})\mu + \gamma_{12}\gamma_{14}$$

substituindo Δ_s , obtemos

$$\begin{aligned}\Delta_w = & D_1(t_5 - u)\Delta_\tau + D_1(t_6 + r_2)\Delta_\theta + \\ & (D_1\nu_2 - Q_1\gamma_{11}\gamma_{13})\mu^2 + (D_1\nu_1 + q_1 - Q_1\gamma_{11}\gamma_{14} - Q_1\gamma_{12}\gamma_{13})\mu + \\ & D_1\nu_0 - w - Q_1\gamma_{12}\gamma_{14}\end{aligned}$$

Isolando Δ_z

A partir da equação (4) :

$$-A_2^T\Delta_y + c_2\Delta_\tau - r_4\Delta_\theta - \Delta_z = 0$$

e substituindo Δ_y obtemos

$$\Delta_z = (c_2 - f_5)\Delta_\tau + (-f_6 - r_4)\Delta_\theta - \sigma_5\mu^2 + (-\sigma_4 - f_7)\mu - \sigma_3 - f_8$$

Isolando Δ_x

A partir da equação (9a):

$$D_3\Delta_x + \Delta_z - \mu q_3 + z + Q_3\delta_x\delta_z = 0$$

e multiplicando-se por $D_5 = D_3^{-1}$ obtemos

$$D_5 (D_5^{-1} \Delta_x + \Delta_z - \mu q_3 + z + Q_3 \delta_x \delta_z) = 0$$

e substituindo Δ_z obtemos

$$\begin{aligned} \Delta_x &= D_5 (f_5 - c_2) \Delta_\tau + D_5 (f_6 + r_4) \Delta_\theta + D_5 (\sigma_5 - Q_3 \gamma_7 \gamma_9) \mu^2 \\ &+ D_5 (\sigma_4 - Q_3 \gamma_8 \gamma_9 + q_3 - Q_3 \gamma_7 \gamma_{10} + f_7) \mu \\ &+ D_5 (f_8 - z + \sigma_3 - Q_3 \gamma_8 \gamma_{10}) \end{aligned}$$

Isolando Δ_h

A partir da equação (8a):

$$D_2 \Delta_v + \Delta_h - \mu q_2 + h + Q_2 \delta_h \delta_v = 0$$

e substituindo Δ_v obtemos

$$\begin{aligned} \Delta_h &= (-Q_2 \gamma_3 \gamma_5 - D_2 \nu_2) \mu^2 \\ &+ (-D_2 \nu_1 + q_2 - Q_2 \gamma_4 \gamma_5 - Q_2 \gamma_3 \gamma_6) \mu \\ &- D_2 t_5 \Delta_\tau - D_2 t_6 \Delta_\theta - h - D_2 \nu_0 - Q_2 \gamma_4 \gamma_6 \end{aligned}$$

Cálculo de Δ_τ , Δ_θ , e Δ_κ em função de μ

A partir das equações (5), (6) e (10) substituindo Δ_y , Δ_w , Δ_v e Δ_x obtemos um sistema de três equações nas variáveis reais Δ_τ , Δ_θ , e Δ_κ , o que nos permite expressá-las em termos de μ e, conseqüentemente, expressar os demais passos em termos de μ .

$$b^T \Delta_y - u^T \Delta_w - c_1^T \Delta_v - c_2^T \Delta_x - r_5 \Delta_\theta - \Delta_\kappa = 0 \quad (5)$$

$$r_1^T \Delta_y + r_2^T \Delta_w + r_3^T \Delta_v + r_4^T \Delta_x + r_5 \Delta_\tau = 0 \quad (6)$$

$$\kappa \Delta_\tau + \tau \Delta_\kappa - \mu + \tau \kappa + \delta_\kappa \delta_\tau = 0 \quad (10)$$

Da equação 5 temos

$$b^T \Delta_y - u^T \Delta_w - c_1^T \Delta_v - c_2^T \Delta_x - r_5 \Delta_\theta - \Delta_\kappa = 0$$

ou

$$\begin{aligned} & (b^T t_1 - c_1^T t_5 - c_2^T (f_5 - c_2) D_5 - u^T (t_5 - u) D_1) \Delta_\tau + \\ & (b^T t_2 - c_2^T (f_6 + r_4) D_5 - u^T (t_6 + r_2) D_1 - r_5 - c_1^T t_6) \Delta_\theta \\ & - \Delta_\kappa + \\ & (b^T \eta_2 - c_2^T (-Q_3 \gamma_7 \gamma_9 + \sigma_5) D_5 - c_1^T \nu_2 - u^T (\nu_2 D_1 - Q_1 \gamma_{11} \gamma_{13})) \mu^2 + \\ & - c_2^T (\sigma_4 - Q_3 \gamma_8 \gamma_9 + q_3 - Q_3 \gamma_7 \gamma_{10} + f_7) D_5 \mu - c_1^T \nu_1 \mu \\ & - u^T (\nu_1 D_1 + q_1 - Q_1 \gamma_{11} \gamma_{14} - Q_1 \gamma_{12} \gamma_{13}) \mu + b^T (\eta_1 + t_3) \mu + \\ & b^T (\eta_0 + t_4) - c_1^T \nu_0 - c_2^T (f_8 - z + \sigma_3 - Q_3 \gamma_8 \gamma_{10}) D_5 - u^T (-Q_1 \gamma_{12} \gamma_{14} + D_1 \nu_0 - w) \\ & = 0 \end{aligned}$$

e definimos

$$\begin{aligned} \omega_1 &= b^T t_1 - c_1^T t_5 - c_2^T (f_5 - c_2) D_5 - u^T (t_5 - u) D_1 \\ \omega_2 &= b^T t_2 - c_1^T t_6 - c_2^T (f_6 + r_4) D_5 - u^T (t_6 + r_2) D_1 - r_5 \end{aligned}$$

$$\begin{aligned}
\xi_1 &= b^T \eta_2 - c_2^T (-Q_3 \gamma_7 \gamma_9 + \sigma_5) D_5 - c_1^T \nu_2 - u^T (\nu_2 D_1 - Q_1 \gamma_{11} \gamma_{13}) \\
\xi_2 &= -c_2^T (\sigma_4 - Q_3 \gamma_8 \gamma_9 + q_3 - Q_3 \gamma_7 \gamma_{10} + f_7) D_5 - c_1^T \nu_1 \\
&\quad - u^T (\nu_1 D_1 + q_1 - Q_1 \gamma_{11} \gamma_{14} - Q_1 \gamma_{12} \gamma_{13}) + b^T (\eta_1 + t_3) \\
\xi_3 &= b^T (\eta_0 + t_4) - c_1^T \nu_0 - c_2^T (f_8 - z + \sigma_3 - Q_3 \gamma_8 \gamma_{10}) D_5 - u^T (-Q_1 \gamma_{12} \gamma_{14} + D_1 \nu_0 - w)
\end{aligned}$$

Da equação 6 temos

$$r_1^T \Delta_y + r_2^T \Delta_w + r_3^T \Delta_v + r_4^T \Delta_x + r_5 \Delta_\tau = 0$$

ou

$$\begin{aligned}
&(r_1^T t_1 + r_4^T (-c_2 + f_5) D_5 + r_3^T t_5 + r_2^T (t_5 - u) D_1 + r_5) \Delta_\tau \\
&+ (r_1^T t_2 + r_4^T (f_6 + r_4) D_5 + r_3^T t_6 + r_2^T (t_6 + r_2) D_1) \Delta_\theta \\
&+ (r_2^T (\nu_2 D_1 - Q_1 \gamma_{11} \gamma_{13}) + r_1^T \eta_2 + r_4^T (-Q_3 \gamma_7 \gamma_9 + \sigma_5) D_5 + r_3^T \nu_2) \mu^2 \\
&+ r_1^T (\eta_1 + t_3) \mu + r_4^T (\sigma_4 - Q_3 \gamma_8 \gamma_9 + q_3 - Q_3 \gamma_7 \gamma_{10} + f_7) D_5 \mu + r_3^T \nu_1 \mu \\
&+ r_2^T (\nu_1 D_1 + q_1 - Q_1 \gamma_{11} \gamma_{14} - Q_1 \gamma_{12} \gamma_{13}) \mu \\
&+ r_1^T (\eta_0 + t_4) + r_4^T (f_8 - z + \sigma_3 - Q_3 \gamma_8 \gamma_{10}) D_5 + r_3^T \nu_0 + r_2^T (-Q_1 \gamma_{12} \gamma_{14} + D_1 \nu_0 - w) \\
&= 0
\end{aligned}$$

e definimos

$$\begin{aligned}
\omega_3 &= r_1^T t_1 + r_4^T (-c_2 + f_5) D_5 + r_3^T t_5 + r_2^T (t_5 - u) D_1 + r_5 \\
\omega_4 &= r_1^T t_2 + r_4^T (f_6 + r_4) D_5 + r_3^T t_6 + r_2^T (t_6 + r_2) D_1 \\
\xi_4 &= r_2^T (\nu_2 D_1 - Q_1 \gamma_{11} \gamma_{13}) + r_1^T \eta_2 + r_4^T (-Q_3 \gamma_7 \gamma_9 + \sigma_5) D_5 + r_3^T \nu_2 \\
\xi_5 &= r_1^T (\eta_1 + t_3) + r_4^T (\sigma_4 - Q_3 \gamma_8 \gamma_9 + q_3 - Q_3 \gamma_7 \gamma_{10} + f_7) D_5 + r_3^T \nu_1 \\
&\quad + r_2^T (\nu_1 D_1 + q_1 - Q_1 \gamma_{11} \gamma_{14} - Q_1 \gamma_{12} \gamma_{13}) \\
\xi_6 &= r_1^T (\eta_0 + t_4) + r_4^T (f_8 - z + \sigma_3 - Q_3 \gamma_8 \gamma_{10}) D_5 + r_3^T \nu_0 + r_2^T (-Q_1 \gamma_{12} \gamma_{14} + D_1 \nu_0 - w)
\end{aligned}$$

Da equação 10 temos

$$\kappa\Delta_\tau + \tau\Delta_\kappa - \mu + \tau\kappa + \delta_\kappa\delta_\tau = 0$$

e definindo

$$\delta_\kappa\delta_\tau = p_5p_1\mu^2 + (p_6p_1 + p_5p_2)\mu + p_6p_2$$

temos

$$\kappa\Delta_\tau + \tau\Delta_\kappa + p_5p_1\mu^2 + (p_6p_1 + p_5p_2 - 1)\mu + p_6p_2 + \tau\kappa = 0$$

obtendo assim o sistema

$$\omega_1\Delta_\tau + \omega_2\Delta_\theta - \Delta_\kappa + \xi_1\mu^2 + \xi_2\mu + \xi_3 = 0$$

$$\omega_3\Delta_\tau + \omega_4\Delta_\theta + \xi_4\mu^2 + \xi_5\mu + \xi_6 = 0$$

$$\kappa\Delta_\tau + \tau\Delta_\kappa + p_5p_1\mu^2 + (p_6p_1 + p_5p_2 - 1)\mu + p_6p_2 + \tau\kappa = 0$$

ou

$$\begin{array}{rccccccc} \omega_1\Delta_\tau & +\omega_2\Delta_\theta & -\Delta_\kappa & = & -\xi_1\mu^2 & & -\xi_2\mu & & -\xi_3 \\ \omega_3\Delta_\tau & +\omega_4\Delta_\theta & & = & -\xi_4\mu^2 & & -\xi_5\mu & & -\xi_6 \\ \kappa\Delta_\tau & & +\tau\Delta_\kappa & = & -p_5p_1\mu^2 & - & (p_6p_1 + p_5p_2 - 1)\mu & - & p_6p_2 - \tau\kappa \end{array}$$

cuja solução é

$$\Delta_\tau = \frac{(\tau\omega_4\xi_1 - \tau\xi_4\omega_2 + \omega_4p_5p_1)\mu^2}{\tau\omega_4\omega_1 - \tau\omega_2\omega_3 + \kappa\omega_4} \frac{(\tau\omega_4\xi_2 + \omega_4p_5p_2 - \tau\xi_5\omega_2 - \omega_4 + \omega_4p_6p_1)\mu}{\tau\omega_4\omega_1 - \tau\omega_2\omega_3 + \kappa\omega_4} \frac{(\omega_4p_6p_2 + \tau\omega_4\xi_3 - \tau\xi_6\omega_2 + \omega_4\tau\kappa)}{\tau\omega_4\omega_1 - \tau\omega_2\omega_3 + \kappa\omega_4}$$

$$\Delta_{\theta} = \frac{\omega_3 p_5 p_1 - \kappa \xi_4 - \xi_4 \tau \omega_1 + \omega_3 \tau \xi_1}{\tau \omega_4 \omega_1 - \tau \omega_2 \omega_3 + \kappa \omega_4} \mu^2$$

$$+ \frac{\omega_3 \tau \xi_2 + \omega_3 p_6 p_1 + \omega_3 p_5 p_2 - \omega_3 - \xi_5 \tau \omega_1 - \kappa \xi_5}{\tau \omega_4 \omega_1 - \tau \omega_2 \omega_3 + \kappa \omega_4} \mu$$

$$+ \frac{\omega_3 p_6 p_2 - \kappa \xi_6 + \omega_3 \tau \xi_3 + \omega_3 \tau \kappa - \xi_6 \tau \omega_1}{\tau \omega_4 \omega_1 - \tau \omega_2 \omega_3 + \kappa \omega_4}$$

$$\Delta_{\kappa} = - \frac{p_5 p_1 \omega_1 \omega_4 - \omega_3 p_5 p_1 \omega_2 + \kappa \xi_4 \omega_2 - \kappa \omega_4 \xi_1}{\tau \omega_4 \omega_1 - \tau \omega_2 \omega_3 + \kappa \omega_4} \mu^2$$

$$- \frac{-\omega_3 p_5 p_2 \omega_2 - \omega_1 \omega_4 + p_5 p_2 \omega_1 \omega_4 + \kappa \xi_5 \omega_2 + p_6 p_1 \omega_1 \omega_4 + \omega_3 \omega_2 - \kappa \omega_4 \xi_2 - \omega_3 p_6 p_1 \omega_2}{\tau \omega_4 \omega_1 - \tau \omega_2 \omega_3 + \kappa \omega_4} \mu$$

$$- \frac{-\omega_3 \tau \kappa \omega_2 + p_6 p_2 \omega_1 \omega_4 - \kappa \omega_4 \xi_3 + \tau \kappa \omega_1 \omega_4 + \kappa \xi_6 \omega_2 - \omega_3 p_6 p_2 \omega_2}{\tau \omega_4 \omega_1 - \tau \omega_2 \omega_3 + \kappa \omega_4}$$

Se substituírmos agora

$$t_j = \tau \omega_4 \omega_1 - \tau \omega_2 \omega_3 + \kappa \omega_4$$

$$j_1 = - \frac{(\tau \omega_4 \xi_1 - \tau \xi_4 \omega_2 + \omega_4 p_5 p_1)}{\tau \omega_4 \omega_1 - \tau \omega_2 \omega_3 + \kappa \omega_4}$$

$$j_2 = - \frac{(\tau \omega_4 \xi_2 + \omega_4 p_5 p_2 - \tau \xi_5 \omega_2 - \omega_4 + \omega_4 p_6 p_1)}{\tau \omega_4 \omega_1 - \tau \omega_2 \omega_3 + \kappa \omega_4}$$

$$j_3 = - \frac{(\omega_4 p_6 p_2 + \tau \omega_4 \xi_3 - \tau \xi_6 \omega_2 + \omega_4 \tau \kappa)}{\tau \omega_4 \omega_1 - \tau \omega_2 \omega_3 + \kappa \omega_4}$$

$$j_4 = \frac{\omega_3 p_5 p_1 - \kappa \xi_4 - \xi_4 \tau \omega_1 + \omega_3 \tau \xi_1}{\tau \omega_4 \omega_1 - \tau \omega_2 \omega_3 + \kappa \omega_4}$$

$$j_5 = \frac{\omega_3 \tau \xi_2 + \omega_3 p_6 p_1 + \omega_3 p_5 p_2 - \omega_3 - \xi_5 \tau \omega_1 - \kappa \xi_5}{\tau \omega_4 \omega_1 - \tau \omega_2 \omega_3 + \kappa \omega_4}$$

$$j_6 = \frac{\omega_3 p_6 p_2 - \kappa \xi_6 + \omega_3 \tau \xi_3 + \omega_3 \tau \kappa - \xi_6 \tau \omega_1}{\tau \omega_4 \omega_1 - \tau \omega_2 \omega_3 + \kappa \omega_4}$$

$$j_7 = \frac{p_5 p_1 \omega_1 \omega_4 - \omega_3 p_5 p_1 \omega_2 + \kappa \xi_4 \omega_2 - \kappa \omega_4 \xi_1}{\tau \omega_4 \omega_1 - \tau \omega_2 \omega_3 + \kappa \omega_4}$$

$$j_8 = \frac{-\omega_3 p_5 p_2 \omega_2 - \omega_1 \omega_4 + p_5 p_2 \omega_1 \omega_4 + \kappa \xi_5 \omega_2 + p_6 p_1 \omega_1 \omega_4 + \omega_3 \omega_2 - \kappa \omega_4 \xi_2 - \omega_3 p_6 p_1 \omega_2}{\tau \omega_4 \omega_1 - \tau \omega_2 \omega_3 + \kappa \omega_4}$$

$$j_9 = \frac{-\omega_3 \tau \kappa \omega_2 + p_6 p_2 \omega_1 \omega_4 - \kappa \omega_4 \xi_3 + \tau \kappa \omega_1 \omega_4 + \kappa \xi_6 \omega_2 - \omega_3 p_6 p_2 \omega_2}{\tau \omega_4 \omega_1 - \tau \omega_2 \omega_3 + \kappa \omega_4}$$

então

$$\Delta_\tau = j_1 \mu^2 + j_2 \mu + j_3$$

$$\Delta_\theta = j_4 \mu^2 + j_5 \mu + j_6$$

$$\Delta_\kappa = j_7 \mu^2 + j_8 \mu + j_9$$

Cálculo de $\Delta_v, \Delta_h, \Delta_x, \Delta_z, \Delta_w$ e Δ_s em função de μ

Devemos substituir os valores que acabamos de obter de $\Delta_\tau, \Delta_\theta$, e Δ_κ nas equações que definem $\Delta_y, \Delta_v, \Delta_h, \Delta_x, \Delta_z, \Delta_w, \Delta_s$ isto é,

$$\begin{aligned}
 \Delta_y &= (\eta_2 + t_2 j_4 + t_1 j_1) \mu^2 + (t_1 j_2 + t_2 j_5 + \eta_1 + t_3) \mu \\
 &\quad + t_1 j_3 + \eta_0 + t_2 j_6 + t_4 \\
 \Delta_v &= (t_5 j_1 + t_6 j_4 + \nu_2) \mu^2 + (t_5 j_2 + \nu_1 + t_6 j_5) \mu + t_5 j_3 + t_6 j_6 + \nu_0 \\
 \Delta_h &= D_2 (-\nu_2 - t_6 j_4 - t_5 j_1) \mu^2 \\
 &\quad + (D_2 (-t_5 j_2 - \nu_1 - t_6 j_5) + q_2) \mu \\
 &\quad + D_2 (-\nu_0 - t_6 j_6 - t_5 j_3) - h \\
 \Delta_x &= D_5 ((-c_2 + f_5) j_1 + (f_6 + r_4) j_4 - Q_3 \gamma_7 \gamma_9 + \sigma_5) \mu^2 \\
 &\quad + D_5 ((-c_2 + f_5) j_2 + (f_6 + r_4) j_5 + \sigma_4 - Q_3 \gamma_8 \gamma_9 + q_3 - Q_3 \gamma_7 \gamma_{10} + f_7) \mu \\
 &\quad + D_5 ((-c_2 + f_5) j_3 - z + \sigma_3 + (f_6 + r_4) j_6 + f_8 - Q_3 \gamma_8 \gamma_{10}) \\
 \Delta_z &= ((c_2 - f_5) j_1 - \sigma_5 + (-f_6 - r_4) j_4) \mu^2 \\
 &\quad + ((c_2 - f_5) j_2 - \sigma_4 - f_7 + (-f_6 - r_4) j_5) \mu \\
 &\quad + (c_2 - f_5) j_3 - f_8 + (-f_6 - r_4) j_6 - \sigma_3 \\
 \Delta_w &= (-Q_1 \gamma_{11} \gamma_{13} + D_1 (\nu_2 + (t_6 + r_2) j_4 + (t_5 - u) j_1)) \mu^2 \\
 &\quad + (q_1 - Q_1 \gamma_{11} \gamma_{14} - Q_1 \gamma_{12} \gamma_{13} + D_1 ((t_6 + r_2) j_5 + \nu_1 + (t_5 - u) j_2)) \mu \\
 &\quad + D_1 ((t_6 + r_2) j_6 + (t_5 - u) j_3 + \nu_0) - w - Q_1 \gamma_{12} \gamma_{14} \\
 \Delta_s &= ((-t_6 - r_2) j_4 + (u - t_5) j_1 - \nu_2) \mu^2 \\
 &\quad + ((u - t_5) j_2 + (-t_6 - r_2) j_5 - \nu_1) \mu \\
 &\quad + (u - t_5) j_3 - \nu_0 + (-t_6 - r_2) j_6
 \end{aligned}$$

Para simplificar as expressões que se seguem definimos para Δ_y :

$$\begin{aligned}\Delta_y = & (\eta_2 + t_2j_4 + t_1j_1) \mu^2 + (t_1j_2 + t_2j_5 + \eta_1 + t_3) \mu \\ & + t_1j_3 + \eta_0 + t_2j_6 + t_4\end{aligned}$$

$$\phi_1 = \eta_2 + t_2j_4 + t_1j_1$$

$$\phi_2 = t_1j_2 + t_2j_5 + \eta_1 + t_3$$

$$\phi_3 = t_1j_3 + \eta_0 + t_2j_6 + t_4$$

para Δ_v :

$$\Delta_v = (t_5j_1 + t_6j_4 + \nu_2) \mu^2 + (t_5j_2 + \nu_1 + t_6j_5) \mu + t_5j_3 + t_6j_6 + \nu_0$$

$$\phi_4 = t_5j_1 + t_6j_4 + \nu_2$$

$$\phi_5 = t_5j_2 + \nu_1 + t_6j_5$$

$$\phi_6 = t_5j_3 + t_6j_6 + \nu_0$$

para Δ_h :

$$\begin{aligned}\Delta_h = & (-Q_2\gamma_3\gamma_5 - D_2\nu_2 - D_2t_6j_4 - D_2t_5j_1) \mu^2 \\ & + (-D_2t_5j_2 - D_2\nu_1 + q_2 - Q_2\gamma_4\gamma_5 - Q_2\gamma_3\gamma_6 - D_2t_6j_5) \mu \\ & - h - D_2t_6j_6 - D_2t_5j_3 - D_2\nu_0 - Q_2\gamma_4\gamma_6\end{aligned}$$

$$\phi_7 = -Q_2\gamma_3\gamma_5 - D_2\nu_2 - D_2t_6j_4 - D_2t_5j_1$$

$$\phi_8 = -D_2t_5j_2 - D_2\nu_1 + q_2 - Q_2\gamma_4\gamma_5 - Q_2\gamma_3\gamma_6 - D_2t_6j_5$$

$$\phi_9 = -h - D_2t_6j_6 - D_2t_5j_3 - D_2\nu_0 - Q_2\gamma_4\gamma_6$$

para Δ_x :

$$\begin{aligned}\Delta_x &= D_5 ((-c_2 + f_5) j_1 + (f_6 + r_4) j_4 - Q_3 \gamma_7 \gamma_9 + \sigma_5) \mu^2 \\ &\quad + D_5 ((-c_2 + f_5) j_2 + (f_6 + r_4) j_5 + \sigma_4 - Q_3 \gamma_8 \gamma_9 + q_3 - Q_3 \gamma_7 \gamma_{10} + f_7) \mu \\ &\quad + D_5 ((-c_2 + f_5) j_3 - z + \sigma_3 + (f_6 + r_4) j_6 + f_8 - Q_3 \gamma_8 \gamma_{10})\end{aligned}$$

$$\phi_{10} = D_5 ((-c_2 + f_5) j_1 + (f_6 + r_4) j_4 - Q_3 \gamma_7 \gamma_9 + \sigma_5)$$

$$\phi_{11} = D_5 ((-c_2 + f_5) j_2 + (f_6 + r_4) j_5 + \sigma_4 - Q_3 \gamma_8 \gamma_9 + q_3 - Q_3 \gamma_7 \gamma_{10} + f_7)$$

$$\phi_{12} = D_5 ((-c_2 + f_5) j_3 - z + \sigma_3 + (f_6 + r_4) j_6 + f_8 - Q_3 \gamma_8 \gamma_{10})$$

para Δ_z :

$$\begin{aligned}\Delta_z &= ((c_2 - f_5) j_1 - \sigma_5 + (-f_6 - r_4) j_4) \mu^2 \\ &\quad + ((c_2 - f_5) j_2 - \sigma_4 - f_7 + (-f_6 - r_4) j_5) \mu \\ &\quad + (c_2 - f_5) j_3 - f_8 + (-f_6 - r_4) j_6 - \sigma_3\end{aligned}$$

$$\phi_{13} = (c_2 - f_5) j_1 - \sigma_5 + (-f_6 - r_4) j_4$$

$$\phi_{14} = (c_2 - f_5) j_2 - \sigma_4 - f_7 + (-f_6 - r_4) j_5$$

$$\phi_{15} = (c_2 - f_5) j_3 - f_8 + (-f_6 - r_4) j_6 - \sigma_3$$

para Δ_w :

$$\begin{aligned}\Delta_w &= (-Q_1 \gamma_{11} \gamma_{13} + D_1 (\nu_2 + (t_6 + r_2) j_4 + (t_5 - u) j_1)) \mu^2 \\ &\quad + (q_1 - Q_1 \gamma_{11} \gamma_{14} - Q_1 \gamma_{12} \gamma_{13} + D_1 ((t_6 + r_2) j_5 + \nu_1 + (t_5 - u) j_2)) \mu \\ &\quad + D_1 ((t_6 + r_2) j_6 + (t_5 - u) j_3 + \nu_0) - w - Q_1 \gamma_{12} \gamma_{14}\end{aligned}$$

$$\begin{aligned}
\phi_{16} &= -Q_1\gamma_{11}\gamma_{13} + D_1(\nu_2 + (t_6 + r_2)j_4 + (t_5 - u)j_1) \\
\phi_{17} &= q_1 - Q_1\gamma_{11}\gamma_{14} - Q_1\gamma_{12}\gamma_{13} + D_1((t_6 + r_2)j_5 + \nu_1 + (t_5 - u)j_2) \\
\phi_{18} &= D_1((t_6 + r_2)j_6 + (t_5 - u)j_3 + \nu_0) - w - Q_1\gamma_{12}\gamma_{14}
\end{aligned}$$

para Δ_s :

$$\begin{aligned}
\Delta_s &= ((-t_6 - r_2)j_4 + (u - t_5)j_1 - \nu_2)\mu^2 \\
&\quad + ((u - t_5)j_2 + (-t_6 - r_2)j_5 - \nu_1)\mu \\
&\quad + (u - t_5)j_3 - \nu_0 + (-t_6 - r_2)j_6
\end{aligned}$$

$$\begin{aligned}
\phi_{19} &= (-t_6 - r_2)j_4 + (u - t_5)j_1 - \nu_2 \\
\phi_{20} &= (u - t_5)j_2 + (-t_6 - r_2)j_5 - \nu_1 \\
\phi_{21} &= (u - t_5)j_3 - \nu_0 + (-t_6 - r_2)j_6
\end{aligned}$$

de modo que

$$\begin{aligned}
\Delta_y &= \phi_1\mu^2 + \phi_2\mu + \phi_3 \\
\Delta_v &= \phi_4\mu^2 + \phi_5\mu + \phi_6 \\
\Delta_h &= \phi_7\mu^2 + \phi_8\mu + \phi_9 \\
\Delta_x &= \phi_{10}\mu^2 + \phi_{11}\mu + \phi_{12} \\
\Delta_z &= \phi_{13}\mu^2 + \phi_{14}\mu + \phi_{15} \\
\Delta_w &= \phi_{16}\mu^2 + \phi_{17}\mu + \phi_{18} \\
\Delta_s &= \phi_{19}\mu^2 + \phi_{20}\mu + \phi_{21}
\end{aligned}$$

$$\Delta_\tau = j_1\mu^2 + j_2\mu + j_3$$

$$\Delta_\theta = j_4\mu^2 + j_5\mu + j_6$$

$$\Delta_\kappa = j_7\mu^2 + j_8\mu + j_9$$

Cálculo dos coeficientes de φ

Desenvolveremos agora um exemplo do uso dessa parametrização dos passos em termos de μ . Suponhamos que a vizinhança do algoritmo geral seja dada por $\mathcal{N}_2(\beta)$ e para simplificar a notação suporemos que $\alpha = 1$. A condição de que o próximo iterando deve satisfazer nesse caso é

$$\sum_{i=1}^k (\hat{v}_i \hat{h}_i - \mu)^2 + \sum_{i=1}^{n-k} (\hat{x}_i \hat{z}_i - \mu)^2 + \sum_{i=1}^k (\hat{w}_i \hat{s}_i - \mu)^2 + (\hat{\tau} \hat{\kappa} - \mu)^2 - \beta^2 \mu^2 \leq 0$$

Como cada um dos passos pode ser rescrito em termos de suas coordenadas como

$$\Delta_{y,i} = \phi_{1,i}\mu^2 + \phi_{2,i}\mu + \phi_{3,i}$$

$$\Delta_{v,i} = \phi_{4,i}\mu^2 + \phi_{5,i}\mu + \phi_{6,i}$$

$$\Delta_{h,i} = \phi_{7,i}\mu^2 + \phi_{8,i}\mu + \phi_{9,i}$$

$$\Delta_{x,i} = \phi_{10,i}\mu^2 + \phi_{11,i}\mu + \phi_{12,i}$$

$$\Delta_{z,i} = \phi_{13,i}\mu^2 + \phi_{14,i}\mu + \phi_{15,i}$$

$$\Delta_{w,i} = \phi_{16,i}\mu^2 + \phi_{17,i}\mu + \phi_{18,i}$$

$$\Delta_{s,i} = \phi_{19,i}\mu^2 + \phi_{20,i}\mu + \phi_{21,i}$$

$$\Delta_\tau = j_1\mu^2 + j_2\mu + j_3$$

$$\Delta_\theta = j_4\mu^2 + j_5\mu + j_6$$

$$\Delta_\kappa = j_7\mu^2 + j_8\mu + j_9$$

essa condição equivale a

$$\begin{aligned} \varphi(\mu) &= \sum_{i=1}^k ((v_i + \Delta_{v,i})(h_i + \Delta_{h,i}) - \mu)^2 + \sum_{i=1}^{n-k} ((x_i + \Delta_{x,i})(z_i + \Delta_{z,i}) - \mu)^2 + \\ &\quad \sum_{i=1}^k ((w_i + \Delta_{w,i})(s_i + \Delta_{s,i}) - \mu)^2 + ((\tau + \Delta_{\tau})(\kappa + \Delta_{\kappa}) - \mu)^2 - \beta^2 \mu^2 \\ &\leq 0 \end{aligned}$$

onde $\varphi(\mu)$ é um polinômio real em μ cujos coeficientes devemos calcular.

Expandindo cada termo e definindo

$$\begin{aligned} \pi_{1,i} &= (\phi_{5,i}\phi_{7,i} + \phi_{4,i}\phi_{8,i}) \\ \pi_{2,i} &= (v_i + \phi_{6,i}) \\ \pi_{3,i} &= (h_i + \phi_{9,i}) \\ \pi_{4,i} &= (\pi_{2,i}\phi_{7,i} + \phi_{5,i}\phi_{8,i} + \phi_{4,i}\pi_{3,i}) \\ \pi_{5,i} &= (\pi_{2,i}\phi_{8,i} + \phi_{5,i}\pi_{3,i} - 1) \end{aligned}$$

obtemos

$$\begin{aligned}
((v_i + \Delta_{v,i})(h_i + \Delta_{h,i}) - \mu)^2 = & \\
& \phi_{4,i}^2 \phi_{7,i}^2 \mu^8 + \\
& 2\pi_{1,i} \phi_{4,i} \phi_{7,i} \mu^7 + \\
& (2\pi_{4,i} \phi_{4,i} \phi_{7,i} + \pi_{1,i}^2) \mu^6 + \\
& (2\pi_{5,i} \phi_{4,i} \phi_{7,i} + 2\pi_{4,i} \pi_{1,i}) \mu^5 + \\
& (2\pi_{2,i} \pi_{3,i} \phi_{4,i} \phi_{7,i} + 2\pi_{5,i} \pi_{1,i} + \pi_{4,i}^2) \mu^4 + \\
& (2\pi_{2,i} \pi_{3,i} \pi_{1,i} + 2\pi_{5,i} \pi_{4,i}) \mu^3 + \\
& (2\pi_{2,i} \pi_{3,i} \pi_{4,i} + \pi_{5,i}^2) \mu^2 + \\
& 2\pi_{2,i} \pi_{3,i} \pi_{5,i} \mu + \\
& \pi_{2,i}^2 \pi_{3,i}^2
\end{aligned}$$

Definindo

$$\begin{aligned}
\pi_{6,i} &= (\phi_{17,i} \phi_{19,i} + \phi_{16,i} \phi_{20,i}) \\
\pi_{7,i} &= (w_i + \phi_{18,i}) \\
\pi_{8,i} &= (s_i + \phi_{21,i}) \\
\pi_{9,i} &= (\pi_{7,i} \phi_{19,i} + \phi_{17,i} \phi_{20,i} + \phi_{16,i} \pi_{8,i}) \\
\pi_{10,i} &= (\pi_{7,i} \phi_{20,i} + \phi_{17,i} \pi_{8,i} - 1)
\end{aligned}$$

obtemos

$$\begin{aligned}
((w_i + \Delta_{w,i})(s_i + \Delta_{s,i}) - \mu)^2 = & \\
& \phi_{16,i}^2 \phi_{19,i}^2 \mu^8 + \\
& 2\pi_{6,i} \phi_{16,i} \phi_{19,i} \mu^7 + \\
& (2\pi_{9,i} \phi_{16,i} \phi_{19,i} + \pi_{6,i}^2) \mu^6 + \\
& (2\pi_{10,i} \phi_{16,i} \phi_{19,i} + 2\pi_{9,i} \pi_{6,i}) \mu^5 + \\
& (2\pi_{7,i} \pi_{8,i} \phi_{16,i} \phi_{19,i} + 2\pi_{10,i} \pi_{6,i} + \pi_{9,i}^2) \mu^4 + \\
& (2\pi_{7,i} \pi_{8,i} \pi_{6,i} + 2\pi_{10,i} \pi_{9,i}) \mu^3 + \\
& (2\pi_{7,i} \pi_{8,i} \pi_{9,i} + \pi_{10,i}^2) \mu^2 + \\
& 2\pi_{7,i} \pi_{8,i} \pi_{10,i} \mu + \\
& \pi_{7,i}^2 \pi_{8,i}^2
\end{aligned}$$

substituindo

$$\begin{aligned}
\pi_{11,i} &= (\phi_{11,i} \phi_{13,i} + \phi_{10,i} \phi_{14,i}) \\
\pi_{12,i} &= (x_i + \phi_{12,i}) \\
\pi_{13,i} &= (z_i + \phi_{15,i}) \\
\pi_{14,i} &= (\pi_{12,i} \phi_{13,i} + \phi_{11,i} \phi_{14,i} + \phi_{10,i} \pi_{13,i}) \\
\pi_{15,i} &= (\pi_{12,i} \phi_{14,i} + \phi_{11,i} \pi_{13,i} - 1)
\end{aligned}$$

obtemos

$$\begin{aligned}
((x_i + \Delta_{x,i})(z_i + \Delta_{z,i}) - \mu)^2 = & \\
& \phi_{10,i}^2 \phi_{13,i}^2 \mu^8 + \\
& 2\pi_{11,i} \phi_{10,i} \phi_{13,i} \mu^7 + \\
& (2\pi_{14,i} \phi_{10,i} \phi_{13,i} + \pi_{11,i}^2) \mu^6 + \\
& (2\pi_{15,i} \phi_{10,i} \phi_{13,i} + 2\pi_{14,i} \pi_{11,i}) \mu^5 + \\
& (2\pi_{12,i} \pi_{13,i} \phi_{10,i} \phi_{13,i} + 2\pi_{15,i} \pi_{11,i} + \pi_{14,i}^2) \mu^4 + \\
& (2\pi_{12,i} \pi_{13,i} \pi_{11,i} + 2\pi_{15,i} \pi_{14,i}) \mu^3 + \\
& (2\pi_{12,i} \pi_{13,i} \pi_{14,i} + \pi_{15,i}^2) \mu^2 + \\
& 2\pi_{12,i} \pi_{13,i} \pi_{15,i} \mu + \\
& \pi_{12,i}^2 \pi_{13,i}^2
\end{aligned}$$

definindo

$$\begin{aligned}
\pi_{16} &= (j_2 j_7 + j_1 j_8) \\
\pi_{17} &= (\tau + j_3) \\
\pi_{18} &= (\kappa + j_9) \\
\pi_{19} &= (\pi_{17} j_7 + j_2 j_8 + j_1 \pi_{18}) \\
\pi_{20} &= (\pi_{17} j_8 + j_2 \pi_{18} - 1)
\end{aligned}$$

obtemos

$$\begin{aligned} ((\tau + \Delta_\tau)(\kappa + \Delta_\kappa) - \mu)^2 - \beta^2 \mu^2 = & \\ & j_1^2 j_7^2 \mu^8 + \\ & 2\pi_{16} j_1 j_7 \mu^7 + \\ & (2\pi_{19} j_1 j_7 + \pi_{16}^2) \mu^6 + \\ & (2\pi_{20} j_1 j_7 + 2\pi_{19} \pi_{16}) \mu^5 + \\ & (2\pi_{17} \pi_{18} j_1 j_7 + 2\pi_{20} \pi_{16} + \pi_{19}^2) \mu^4 + \\ & (2\pi_{17} \pi_{18} \pi_{16} + 2\pi_{20} \pi_{19}) \mu^3 + \\ & (2\pi_{17} \pi_{18} \pi_{19} + \pi_{20}^2 - \beta^2) \mu^2 + \\ & 2\pi_{17} \pi_{18} \pi_{20} \mu + \\ & \pi_{17}^2 \pi_{18}^2 \end{aligned}$$

Podemos agora calcular

$$\begin{aligned}
\zeta_8 &= \sum_{i=1}^k \phi_{4,i}^2 \phi_{7,i}^2 + \sum_{i=1}^k \phi_{16,i}^2 \phi_{19,i}^2 + \sum_{i=1}^{n-k} \phi_{10,i}^2 \phi_{13,i}^2 + j_1^2 j_7^2 \\
\zeta_7 &= \sum_{i=1}^k 2\pi_{1,i} \phi_{4,i} \phi_{7,i} + \sum_{i=1}^k 2\pi_{6,i} \phi_{16,i} \phi_{19,i} + \sum_{i=1}^{n-k} 2\pi_{11,i} \phi_{10,i} \phi_{13,i} + 2\pi_{16} j_1 j_7 \\
\zeta_6 &= \sum_{i=1}^k (2\pi_{4,i} \phi_{4,i} \phi_{7,i} + \pi_{1,i}^2) + \sum_{i=1}^k (2\pi_{9,i} \phi_{16,i} \phi_{19,i} + \pi_{6,i}^2) + \\
&\quad \sum_{i=1}^{n-k} (2\pi_{14,i} \phi_{10,i} \phi_{13,i} + \pi_{11,i}^2) + (2\pi_{19} j_1 j_7 + \pi_{16}^2) \\
\zeta_5 &= \sum_{i=1}^k (2\pi_{5,i} \phi_{4,i} \phi_{7,i} + 2\pi_{4,i} \pi_{1,i}) + \sum_{i=1}^k (2\pi_{10,i} \phi_{16,i} \phi_{19,i} + 2\pi_{9,i} \pi_{6,i}) + \\
&\quad \sum_{i=1}^{n-k} (2\pi_{15,i} \phi_{10,i} \phi_{13,i} + 2\pi_{14,i} \pi_{11,i}) + (2\pi_{20} j_1 j_7 + 2\pi_{19} \pi_{16}) \\
\zeta_4 &= \sum_{i=1}^k (2\pi_{2,i} \pi_{3,i} \phi_{4,i} \phi_{7,i} + 2\pi_{5,i} \pi_{1,i} + \pi_{4,i}^2) + \\
&\quad \sum_{i=1}^k (2\pi_{7,i} \pi_{8,i} \phi_{16,i} \phi_{19,i} + 2\pi_{10,i} \pi_{6,i} + \pi_{9,i}^2) + \\
&\quad \sum_{i=1}^{n-k} (2\pi_{12,i} \pi_{13,i} \phi_{10,i} \phi_{13,i} + 2\pi_{15,i} \pi_{11,i} + \pi_{14,i}^2) + \\
&\quad (2\pi_{17} \pi_{18} j_1 j_7 + 2\pi_{20} \pi_{16} + \pi_{19}^2) \\
\zeta_3 &= \sum_{i=1}^k (2\pi_{2,i} \pi_{3,i} \pi_{1,i} + 2\pi_{5,i} \pi_{4,i}) + \sum_{i=1}^k (2\pi_{7,i} \pi_{8,i} \pi_{6,i} + 2\pi_{10,i} \pi_{9,i}) + \\
&\quad \sum_{i=1}^{n-k} (2\pi_{12,i} \pi_{13,i} \pi_{11,i} + 2\pi_{15,i} \pi_{14,i}) + (2\pi_{17} \pi_{18} \pi_{16} + 2\pi_{20} \pi_{19})
\end{aligned}$$

$$\begin{aligned}
\zeta_2 &= \sum_{i=1}^k (2\pi_{2,i}\pi_{3,i}\pi_{4,i} + \pi_{5,i}^2) + \sum_{i=1}^k (2\pi_{7,i}\pi_{8,i}\pi_{9,i} + \pi_{10,i}^2) + \\
&\quad \sum_{i=1}^{n-k} (2\pi_{12,i}\pi_{13,i}\pi_{14,i} + \pi_{15,i}^2) + (2\pi_{17}\pi_{18}\pi_{19} + \pi_{20}^2 - \beta^2) \\
\zeta_1 &= \sum_{i=1}^k 2\pi_{2,i}\pi_{3,i}\pi_{5,i} + \sum_{i=1}^k 2\pi_{7,i}\pi_{8,i}\pi_{10,i} + \sum_{i=1}^{n-k} 2\pi_{12,i}\pi_{13,i}\pi_{15,i} + 2\pi_{17}\pi_{18}\pi_{20} \\
\zeta_0 &= \sum_{i=1}^k \pi_{2,i}^2\pi_{3,i}^2 + \sum_{i=1}^k \pi_{7,i}^2\pi_{8,i}^2 + \sum_{i=1}^{n-k} \pi_{12,i}^2\pi_{13,i}^2 + \pi_{17}^2\pi_{18}^2
\end{aligned}$$

e assim

$$\varphi(\mu) = \sum_{i=0}^8 \zeta_i \mu^i$$

Bibliografia

- [1] I. Adler, M. Resende, G. Veiga, e N. Karmakar. An implementation of Karmakar's algorithm for linear programming. *Mathematical Programming*, 44:297–335, 1989.
- [2] D.A. Bayer e J.C. Lagarias. The nonlinear geometry of linear programming. I: Affine and projective scaling trajectories. II: Legendre transform coordinates and central trajectories. *Trans. Amer. Math. Soc.*, 314:499–581, 1989.
- [3] M.S. Bazaraa e J.J. Jarvis. *Linear Programming and Network Flows*. John Wiley & Sons, 1977.
- [4] J. Czyzyc, S. Mehrotra, M. Wagner, e S.J. Wright. *PCx User Guide (Version 1.1)*. Optimization Technology Center, 1997.
- [5] A. V. Fiacco e G. P. McCormick. *Nonlinear Programming : Sequential Unconstrained Minimization Techniques*. John Wiley & Sons, New York, 1968. Reprint : Volume 4 of *SIAM Classics in Applied Mathematics*, SIAM Publications, Philadelphia, PA 19104–2688, USA, 1990.
- [6] K. R. Frisch. The logarithmic potential method for convex programming. Unpublished manuscript, Institute of Economics, University of Oslo, Oslo, Norway, May 1955.
- [7] K. R. Frisch. The multiplex method for linear programming. *Sankhya*, 18:329–362, 1957.

- [8] P.E. Gill, W. Murray, M.A. Saunders, J.A. Tomlin, e M.H. Wright. On projected Newton barrier methods for linear programming and an equivalence to Karmakar's projective method. *Mathematical Programming*, 36:183–209, 1986.
- [9] D. Goldfarb e S. Mehotra. A relaxed version of Karmakar's method. *Mathematical Programming*, 40:289–315, 1988.
- [10] A.J. Goldman e A.W. Tucker. Polyedral convex cones. In H.W. Kuhn e A.W. Tucker, editores, *Linear Inequalities and Related Systems*, páginas 19–40. Princeton University Press, Princeton, NJ, 1956.
- [11] J. Gondzio. Multiple centrality corrections in a primal-dual method for linear programming. *Computational Optimization and Applications*, 6:137–156, 1996.
- [12] C.C. Gonzaga. Path-following methods for linear programming. *SIAM Review*, 34-2:167–224, 1992.
- [13] A. J. Hoffmann, M. Mannos, D. Sokolowsky, e N. Wiegmann. Computational experience in solving linear programs. *Journal of the Society for Industrial and Applied Mathematics (SIAM)*, 1:17–33, 1953.
- [14] P. Huard. Resolution of mathematical programming with nonlinear constraints by the method of centers. In J. Abadie, editor, *Nonlinear Programming*, páginas 207–219. North Holland, Amsterdam, The Netherlands, 1967.
- [15] P.F. Hung e Y. Ye. An asymptotical $\mathcal{O}(\sqrt{n}L)$ -iteration path-following linear programming algorithm that uses wide neighbourhoods. Technical report, Department of Mathematics, Tunghai University, Taichung, Taiwan, March 1994.
- [16] B. Jansen, C. Roos, T. Terlaky, e J.-Ph. Vial. Primal-dual algorithms for linear programming based on the logarithmic barrier methods. *Journal of Optimization Theory and Applications*, 83:1–26, 1994.

- [17] N. K. Karmarkar. A new polynomial-time algorithm for linear programming. *Combinatorica*, 4:373–395, 1984.
- [18] L.G. Khachiyan. A polynomial algorithm in linear programming. *Soviet Mathematics Doklady*, 20:191–194, 1979.
- [19] L.G. Khachiyan. Polynomial algorithms in linear programming. *USSR Computational Mathematics and Mathematical Physics*, 20:53–72, 1980.
- [20] M. Kojima, S. Mizuno, e A. Yoshise. A primal-dual interior point algorithm for linear programming. In N. Megiddo, editor, *Progress in Mathematical Programming : Interior Point and Related Methods*, páginas 29–47. Springer Verlag, New York, 1989.
- [21] N. Megiddo. Pathways to the optimal set in linear programming. In N. Megiddo, editor, *Progress in Mathematical Programming : Interior Point and Related Methods*, páginas 131–158. Springer Verlag, New York, 1989. Versão idêntica em : *Proceedings of the 6th Mathematical Programming Symposium of Japan, Nagoya, Japan*, páginas 1–35, 1986.
- [22] S. Mehrotra. On the implementation of a primal-dual interior point method. *SIAM Journal on Optimization*, 2(4):575–601, 1992.
- [23] R. Monteiro, I. Adler, e M. Resende. A polynomial-time primal-dual affine scaling algorithm for linear and convex quadratic programming and its power series extension. *Mathematics of Operations Research*, 15-2:191–213, 1990.
- [24] R.D.C. Monteiro e I. Adler. Interior path following primal-dual algorithms : Part I : Linear programming. *Mathematical Programming*, 44:27–41, 1989.
- [25] J. von Neumann. On a maximization problem. Manuscript, Institute for Advanced Studies, Princeton University, Princeton, NJ 08544, USA, 1947.

- [26] E. Ng e B.W. Peyton. Block sparse Cholesky algorithms on advanced uniprocessor computers. *SIAM Journal on Scientific Computing*, 14:1034–1056, 1993.
- [27] J. Peng, C. Roos, e T. Terlaky. New complexity analysis of the primal-dual newton method for linear optimization. Technical Report 98-05, Faculty of Technical Mathematics and Informatics, Delft University of Technology, 1998.
- [28] J. Peng, C. Roos, e T. Terlaky. A new class of polynomial primal-dual methods for linear and semidefinite optimization. Technical report, Faculty of Information Technology and Systems, Delft University of Technology, December 1999.
- [29] N. Shor. Utilization of the operation of space dilation in the minimization of convex functions. *Cybernetics*, 6:7–15, 1970.
- [30] G. Sonnevend. An “analytic center” for polyhedrons and new classes of global algorithms for linear (smooth, convex) programming. In A. Prekopa, J. Szelezsan, e B. Strazicky, editores, *System Modelling and Optimization : Proceedings of the 12th IFIP-Conference held in Budapest, Hungary, September 1985*, volume 84 of *Lecture Notes in Control and Information Sciences*, páginas 866–876. Springer Verlag, Berlin, West-Germany, 1986.
- [31] J.M. Todd e B.P. Burrell. An extension of Karmakar’s algorithm for linear programming using dual variables. *Algorithmica*, 1:409–424, 1986.
- [32] C. B. Tompkins. Projection methods in calculation. In H. A. Antosiewicz, editor, *Proceedings of the Second Symposium in Linear Programming*, páginas 425–448, U. S. Air Force, Washington, DC, USA, 1955.
- [33] C. B. Tompkins. Some methods of computational attack on programming problems, other than the simplex method. *Naval Research Logistics Quarterly*, 4:95–96, 1957.

- [34] A.W. Tucker. Dual systems of homogeneous linear relations. In H.W. Kuhn e A.W. Tucker, editores, *Linear Inequalities and Related Systems*, páginas 3–18. Princeton University Press, Princeton, NJ, 1956.
- [35] F.R. VillasBôas. Projeção direta de vetores. Dissertação de Mestrado, IMECC - Universidade Estadual de Campinas, Março 1995.
- [36] X. Xu. An $\mathcal{O}(\sqrt{n}L)$ -iteration large-step infeasible path-following algorithm for linear programming. Technical report, Department of Management Sciences, The University of Iowa, Iowa City, Iowa 52242, USA, 1994.
- [37] X. Xu, P.F. Hung, e Y. Ye. A simplified homogeneous and self-dual linear programming algorithm and its implementation. Technical report, College of Business Administration, The University of Iowa, Iowa City, Iowa 52242, USA, 1993.
- [38] X. Xu e Y. Ye. A generalized homogeneous and self-dual algorithm for linear programming. Technical Report 94-3, College of Business Administration, The University of Iowa, Iowa City, Iowa 52242, USA, 1994.
- [39] Y. Ye, M.J. Todd, e S. Mizuno. An $\mathcal{O}(\sqrt{n}L)$ -iteration homogeneous and self-dual linear programming algorithm. *Mathematics of Operations Research*, 19:53–67, 1994.
- [40] Yinyu Ye. *Interior Point Algorithms: Theory and Analysis*. John Wiley and Sons, New York, 1997.
- [41] D. Yudin e A. Nemirovskii. Informational complexity and efficient methods for the solution of convex extremal problems. *Matekon*, 13-2:3–25, 1976.